HiABP: Hierarchical Initialized ABP for Unsupervised Representation Learning

Jiankai Sun,[1,2] **Rui Liu,**[2] **Bolei Zhou**[1,2]
[1] Centre for Perceptual and Interactive Intelligence
[2] The Chinese University of Hong Kong
{sj019, bzhou}@ie.cuhk.edu.hk, ruiliu@link.cuhk.edu.hk

Abstract

Although Markov chain Monte Carlo (MCMC) is useful for generating samples from the posterior distribution, it often suffers from intractability when dealing with large-scale datasets. To address this issue, we propose Hierarchical Initialized Alternating Back-propagation (HiABP) for efficient Bayesian inference. Especially, we endow Alternating Back-propagation (ABP) method with a well-designed initializer and hierarchical structure, composing the pipeline of Initializing, Improving, and Learning back-propagation. It saves much time for the generative model to initialize the latent variable by constraining a sampler to be close to the true posterior distribution. The initialized latent variable is then improved significantly by an MCMC sampler. Thus the proposed method has the strengths of both methods, i.e., the effectiveness of MCMC and the efficiency of variational inference. Experimental results validate our framework can outperform other popular deep generative models in modeling natural images and learning from incomplete data. We further demonstrate the unsupervised disentanglement of hierarchical latent representation with controllable image synthesis.

Introduction

Alternating Back-propagation (ABP) (Nijkamp et al. 2019; Han et al. 2017) is a newly-introduced generative model that learns a generator mapping latent variables to observations. It performs an EM-like algorithm that infers latent variable and updates generator parameters alternately, following the tradition of alternating operations in unsupervised learning, such as alternating linear regression in the EM algorithm for factor analysis, alternating least squares algorithm for matrix factorization (Kim and Park 2008), and alternating gradient descent algorithm for sparse coding (Olshausen and Field 1997). As a powerful tool which is asymptotically exact to infer latent variables based on the joint posterior distribution of both the latent variables and observations, Markov chain Monte Carlo (MCMC) methods such as Langevin dynamics or Hamiltonian Monte Carlo (HMC) (Neal 2012) are typically adopted for inference. However, MCMC sampling takes a great amount of time to converge, thus making it difficult to cope with large-scale training data.

An alternative to MCMC sampling is Variational Inference (VI) (Bishop 2006), which is typically more computationally efficient. Variational Auto-encoder (VAE) (Kingma and Welling 2014), a type of generative model adopting VI strategy, proves that the cost of VI can be amortized with an inference model and thus could quickly approximate the posterior over the local latent variables. Despite the success of VAE, the improved speed comes at many significant costs, such as the inability of the approximation family to capture the true posterior and the asymmetry of the true distribution under penalization of the KL divergence with too-light tails.

In this paper, we explore the possibility of marrying the high efficiency of the inference network with the exact approximation of ABP method in generative modeling. To achieve this, we propose to take an inference model as an initializer for ABP method, dubbed initialized alternating back-propagation (IABP). The pipeline is composed of *Initializing*, *Improving* and *Learning* back-propagation, which iterates the following three steps: (1) Initializing latent variable for each training example by parameterized inference model; (2) Improving the continuous latent factors by Langevin dynamics; and (3) Learning both inference model and generator given the improved latent factors. The parameter update in the first step is typically performed using Maximum Likelihood estimation (MLE) in unsupervised learning. The MLE provides only a point estimate of the fitted model parameters, while the second step recovers the entire posterior distribution of the model parameters given the data by Langevin dynamics, providing additional information such as parameter uncertainty and correlations. Since the inference time in the first step is negligible compared to the Langevin search time, our model improves the efficiency of ABP model to a great extent. In addition, as the iteration of updating inference model goes, Langevin dynamics obtains a better starting point, which improves its quality as well. Thus we seamlessly combine the amortized inference model and Langevin strategy in our proposed IABP, which makes all the sub-components cooperate in a harmonious manner and thus promotes each other well.

To sum up, this work makes the following contributions:

1 We introduce a novel framework Initialized Alternating Back-propagation (IABP), which consists of *Initializing, Improving* and *Learning* back-propagation. Our framework integrates both fast thinking (through q_ϕ) and slow

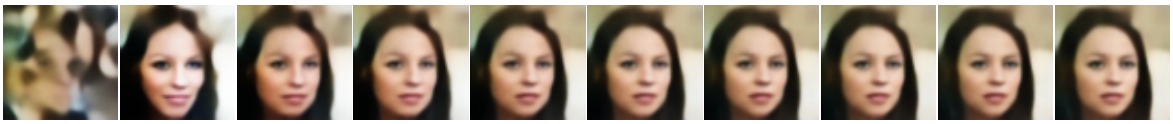
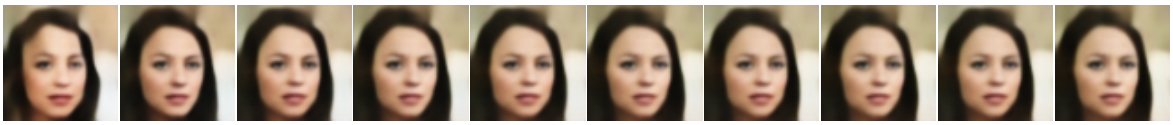
STEP T	1	2	3	4	5	6	7	8	9	10
MSE	0.783	0.120	0.054	0.042	0.038	0.033	0.030	0.029	0.028	0.028
ABP (NON-PERSISTENT)										
MSE	0.074	0.044	0.036	0.030	0.028	0.025	0.025	0.025	0.025	0.025
IABP (OURS)										

Figure 1: **Informative initialization** in IABP can dramatically reduce the number of Langevin steps T needed for convergent Maximum Likelihood (ML) learning. MSE: Mean Square Error between the actual observations and the values predicted by the model.

thinking (through ABP inference). It also combines the bottom-up model (inference model) as well as the top-down model (Langevin dynamics).

2 To verify the scalability of the proposed model for large-scale practical problems, we present that the IABP framework can learn realistic generative models of natural images. It can also learn from incomplete or indirect data.

3 We extend IABP to a hierarchical latent model called Hierarchical Initialized Alternating Back-propagation (HiABP). This model can learn in an unsupervised manner the hierarchical latent representation, by disentangling different layers of latent code.

Related Work

Hybrid Methods: Variants of MCMC and VI Variational auto-encoders (VAEs) (Kingma and Welling 2014) are Deep Gaussian Models trained by using mean-field VI. A more flexible variational family called normalizing flows (NFs) is used by (Rezende and Mohamed 2015) to improve over VAEs. However, the main limitation of VAEs and NFs is the bias present in their variational approximations. This bias can be quite high, even in the case of NFs, since the transformations have to be rather simple to ensure invertibility and to reduce computational costs. MCMC methods are less popular than VI for the reason that they are more computationally expensive. Many recent works seek a better balance between efficiency and bias by combining MCMC and VI. In Hoffman (2017), the gradient of the true likelihood is directly approximated by using Fisher's identity and HMC to obtain approximate samples from $p_\theta(z|x)$. However, the MCMC bias can be significant when one has multimodal latent posteriors and is strongly dependent on both the initial distribution and θ. HVAE (Caterini, Doucet, and Sejdinovic 2018) reduces variational bias by optimizing an ELBO specified in terms of the tractable joint density of short MCMC chains, but the proposed ELBO becomes looser and looser as the chain grows longer since the auxiliary momentum variables are sampled only once at the beginning of the chain, which reduces the empirical performance of HMC. Introduced in (Ruiz and Titsias 2019), Variational Contrastive Divergence (VCD) is a new divergence that replaces the standard Kullback-Leibler (KL) divergence used in VI. The wake-sleep algorithm (Hinton et al. 1995) does not correspond to the optimization of (a bound of) the marginal likelihood. In contrast to these methods, our approach is based on maximum likelihood, which is theoretically the most accurate estimator, following the tradition of alternating operations in unsupervised learning.

Unsupervised Hierarchical Latent Disentanglement. Hierarchical deep generative models (Rezende, Mohamed, and Wierstra 2014) follow hierarchy of latent variables $\boldsymbol{z} = \{\boldsymbol{z}^1, ..., \boldsymbol{z}^L\}$, in addition to the observed variables $\boldsymbol{x}$ is defined as Equation (1) using chain rule:

$$p(\boldsymbol{x}, \boldsymbol{z^1}, ..., \boldsymbol{z^L}) = p(\boldsymbol{x}|\boldsymbol{z}^{>0})\Pi_{l=1}^{L-1} p(\boldsymbol{z}^l|\boldsymbol{z}^{>l})p(\boldsymbol{z}^L), \quad (1)$$

where $\boldsymbol{z}_{>l}$ indicates $\boldsymbol{z}_{l+1}, ..., \boldsymbol{z}_L$. As is a very challenging task, training such a hierarchical deep generative model in an unsupervised manner usually focuses on learning a hierarchy of latent variables by stacking single layer models on top of each other (Sø nderby et al. 2016). Reasonable hierarchical network structure can be, by itself, highly effective at learning disentangled representations.

Method

In the section, we first introduce the preliminary about Alternating Back-propagation and its variants. Then, we describe the design of a particular initializer and Langevin dynamics. Finally, we introduce our generic single stochastic layer IABP framework and the extension of IABP to hierarchical latent variable model HiABP.

Preliminary

Alternating Back-propagation (ABP). In ABP paradigm (Nijkamp et al. 2019; Han et al. 2017; Xie et al. 2019; Han et al. 2019), a generator network is learned by iterating the following two steps: (a) inferring the latent variables by Langevin dynamics that are sampled from the posterior distribution of the latent variables. (b) updating the generator parameters based on the inferred latent

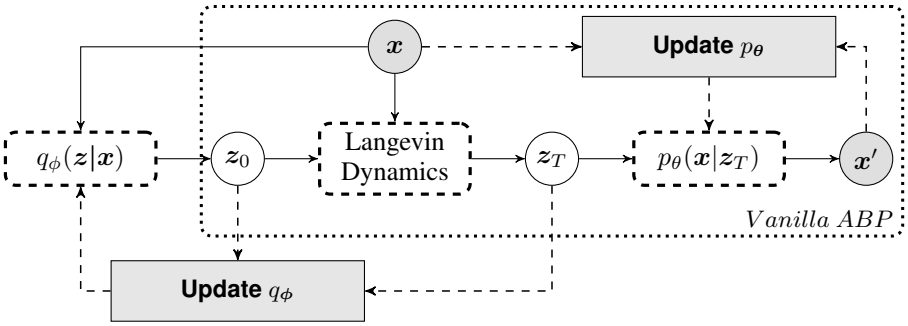


Figure 2: IABP Framework. Different from vanilla ABP ($\boldsymbol{z}_0 \sim \mathcal{N}(0,1)$), IABP iterates the following three steps: (1) parameterized $q_{\boldsymbol{\phi}}$ quickly generates initial latent variable $\boldsymbol{z}_0$; (2) Improving latent variable via T-step Langevin dynamics starting from $\boldsymbol{z}_0$ to obtain $\boldsymbol{z}_T$; (3) Updating $q_{\boldsymbol{\phi}}$ and $p_{\boldsymbol{\theta}}$ (dashed arrow part). $q_{\boldsymbol{\phi}}$ is typically updated using Maximum Likelihood estimation (MLE) in unsupervised learning. The MLE provides only a point estimate of the fitted model parameters, while step (2) recovers the entire posterior distribution of the model parameters given in the data by Langevin dynamics. Since the initialization is improved after each $q_{\boldsymbol{\phi}}$ update, the quality of the Langevin dynamics' samples also improves.

variables. Both steps involve gradient computations based on back-propagation, thus it is called as "alternating back-propagation". Note that in the training stage, in step (a), for observed example $\boldsymbol{x}$, the first round of Langevin starts from random noise $\mathcal{N}(0,1)$. In later rounds of Langevin dynamics, instead of starting from $\mathcal{N}(0,1)$, vanilla ABP starts from the value obtained in the previous round. This is usually called *persistent chain* in the literature. In (Nijkamp et al. 2019), in step (a), they are essentially doing *non-persistent* ABP, meaning that in each round of Langevin dynamics, they always starts from random noise $\mathcal{N}(0,1)$, then doing the finite-step (*e.g.*, 20-step) Langevin updating. Xing *et al.* (Xing et al. 2019; Xing et al. 2020) study the unsupervised disentanglement by extending the ABP model to infer the Langevin dynamics of two groups of independent latent variables for the representation of shape and appearance. The hierarchical compositional model is explored in (Xing et al. 2020), under the engine of ABP, by unifying the top-down generator network and the sparse coding model. Nijkamp *et al.* (Nijkamp et al. 2020) extend the ABP model with the hierarchical latent variables. Recently, a joint training scheme is proposed (Han et al. 2020), where the latent energy-based model (EBM) serves as a critic of the generator model, while the generator model and the inference model in VAE serve as the approximate synthesis sampler and inference sampler of the latent EBM.

Informative Initializer for IABP

Different from the previous work, the step of the Langevin dynamics in our method is initialized from the values of the latent variables produced in $q_{\boldsymbol{\phi}}$ network, which is called *Informative Initializaton* in literature. It is impractical to run long chains to sample from $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. Different from non-persistent initialization starting from noise, or persistent initialization from the value obtained from the previous round, we propose to find an optimal informative initializer $q_{\boldsymbol{\phi}}$ to closely approximate the stationary distribution of the Langevin dynamics.

$$\boldsymbol{z}_0 = \mu_{\boldsymbol{\phi}}(\boldsymbol{x}) + \sigma_{\boldsymbol{\phi}}(\boldsymbol{x}) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0,1). \tag{2}$$

Suppose our inference network $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ with parameter $\boldsymbol{\phi}$ is $\mathcal{N}(\mu_{\boldsymbol{\phi}}(\boldsymbol{x}), \sigma^2_{\boldsymbol{\phi}}(\boldsymbol{x}))$. $\odot$ indicates the element-wise product. $\boldsymbol{z}_0$ is generated as Equation (2) by the current inference model $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ using reparameterization trick for each training observation. Experiments demonstrate that such informative initialization can dramatically reduce the number of Langevin steps needed for convergent ML learning (see Figure 1).

Langevin Dynamics

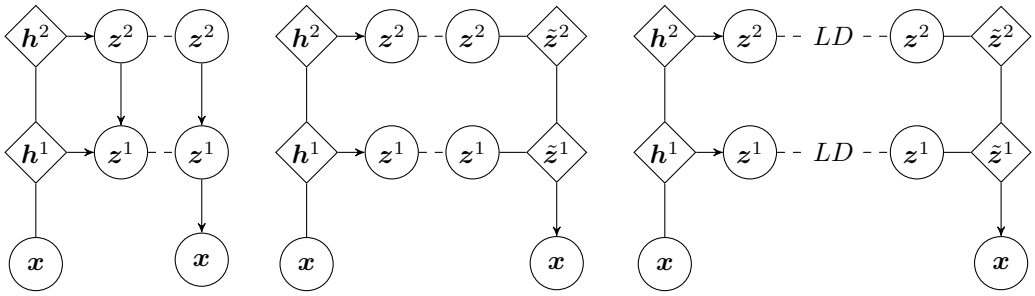
Langevin dynamics (Neal 2012) in Equation (3) is a stochastic sampling counterpart of gradient descent used in our framework for sampling from $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$, which iterates

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \frac{s^2}{2}\frac{\partial}{\partial \boldsymbol{z}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_t|\boldsymbol{x}) + s\boldsymbol{\epsilon}_t, \quad t = 1, ..., T, \tag{3}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0,1)$ is Gaussian noise injected in Langevin sampling, t indexes the time step of Langevin dynamics, s is the Langevin step size, $s\boldsymbol{\epsilon}_t$ is the white noise diffusion term in Langevin dynamics to create randomness for sampling from $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. For small step size s, the marginal distribution of $\boldsymbol{z}_t$ will converge to $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ as $t \to \infty$ regardless of the initial distribution of $\boldsymbol{z}_0$. More specifically, let $q_t(\boldsymbol{z})$ be the marginal distribution of $\boldsymbol{z}_t$ of Langevin dynamics, then $KL(q_t(\boldsymbol{z})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})) \to 0$ monotonically, that is, by increasing t, $KL(q_t(\boldsymbol{z})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}))$ is reduced. $-\log p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ is the gradient descent term consisted in Langevin dynamics. In the case of our generator network $g_{\boldsymbol{\theta}}$, it amounts to the gradients descent on penalized reconstruction error $||\boldsymbol{z}||^2/2 + ||\boldsymbol{x} - g_{\boldsymbol{\theta}}(\boldsymbol{z})||^2/2\sigma^2$, which is the negative log-likelihood of Gaussian distribution. We get improved latent variable $\boldsymbol{z}_T$ after T steps Langevin updates from $\boldsymbol{z}_0$.

Learning Procedure for IABP

We design a simple and easily reproducible pipeline for single-layer IABP, as shown in Figure 2. Let $p_{data}(\boldsymbol{x})$ be the data distribution that generates the example $\boldsymbol{x}$. The learning of parameters $\boldsymbol{\theta}$ of $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ can be based on $\min_{\boldsymbol{\theta}} KL(p_{data}(\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{x}))$. If we observe training examples $\{\boldsymbol{x}_i, i = 1, ..., n\} \sim p_{data}$, the learning procedure



(a) LVAE (b) VLAE (c) Ours: HiABP

Figure 3: Inference and generative models for LVAE (Sø nderby et al. 2016) (left), VLAE (Zhao, Song, and Ermon 2017) (middle) and HiABP (right). Circles indicate stochastic nodes, and squares are deterministically computed nodes. Solid lines with arrows denote conditional probabilities; solid lines without arrows denote deterministic mappings; dash lines indicate regularization to match the prior $p(\boldsymbol{z})$. LD represents Langevin dynamics.

Algorithm 1 Hierarchical Initialized Alternating Back-propgation (HiABP)

Require: Latent layer L, $p_{\boldsymbol{\theta}}$ learning rate $\eta_{\boldsymbol{\theta}}$, $q_{\boldsymbol{\phi}}$ learning rate $\eta_{\boldsymbol{\phi}}$, observed examples $\{\boldsymbol{x}^{(i)}\}_{i=1}^n$, batch size m, number of Langevin steps T, step size s.

Ensure: Weights $\boldsymbol{\theta}, \boldsymbol{\phi}$

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters.

repeat

Draw observed examples $\{\boldsymbol{x}^{(i)}\}_{i=1}^m$.

Initializing Alternating-backpropagation: For $\ell = 1, ..., L, i = 1, ..., m$, draw latent vectors $\boldsymbol{z}_0^{(i),\ell}$ according to Equation (7).

Improving Alternating-backpropagation: $\{\boldsymbol{z}_0^{(i)}\}_{i=1}^m = \{\boldsymbol{z}_0^{(i),0}, \boldsymbol{z}_0^{(i),1}, ..., \boldsymbol{z}_0^{(i),L}\}_{i=1}^m$, infer $\{\boldsymbol{z}_T^{(i)}\}_{i=1}^m$ by T-steps of dynamics in Equation (3) with step size s.

Learning Alternating-backpropagation: Update $\boldsymbol{\theta}$ with learning rate $\eta_{\boldsymbol{\theta}}$ according to Equation (4). Update $\boldsymbol{\phi}$ with learning rate $\eta_{\boldsymbol{\phi}}$ according to Equation (5).

until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$

return $\boldsymbol{\theta}, \boldsymbol{\phi}$

would proceed as the following iteration: (1) Learning generator parameter $\boldsymbol{\theta}$ using $\boldsymbol{z}_T$ by Equation (4). (2) Learning inference net parameter $\boldsymbol{\phi}$ by Equation (5), which encourages $q_{\boldsymbol{\phi}}$ to approximate the desired target distribution.

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}_T^{(i)}|\boldsymbol{x}^{(i)})} \left[\frac{\partial}{\partial \boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{z}_T^{(i)}) \right], \quad (4)$$

$$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \eta_{\boldsymbol{\phi}} \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial \boldsymbol{\phi}} \left[\frac{||\boldsymbol{z}_T^{(i)} - \mu_{\boldsymbol{\phi}}(\boldsymbol{x}^{(i)})||^2}{2\sigma_{\boldsymbol{\phi}}^2(\boldsymbol{x}^{(i)})} + 0.5 \log \sigma_{\boldsymbol{\phi}}^2(\boldsymbol{x}^{(i)}) \right], \quad (5)$$

where $\eta_{\boldsymbol{\theta}}$ is the learning rate for $p_{\boldsymbol{\theta}}$, $\eta_{\boldsymbol{\phi}}$ is the learning rate for $q_{\boldsymbol{\phi}}$. The gradients in both steps are computed by back-propagation. IABP can be regarded as a single-layer HiABP ($L = 1$), of which the learning procedure of is summarized in Algorithm 1.

Note that IABP actually integrates both fast thinking (through $q_{\boldsymbol{\phi}}$) and slow thinking (through ABP inference). It also combines the bottom-up model (inference model) as well as the top-down model (Langevin dynamics). Neither of the previous VAE methods nor vanilla ABP models have this design.

Hierarchical Initialized Alternating-backpropagation (HiABP)

Training the hierarchical latent variable models using MCMC posterior sampling or VAE is challenging. On one hand, the starting point in MCMC posterior sampling is always random (Gaussian or Uniform), which is not informative, and the Gaussian parametrized generator model is not as expressive as neural network approximator in Equation (6) which can implicitly represent many complicated distributions other than Gaussian. On the other hand, VAE uses an encoder parametrized by Gaussian to approximate the intractable posterior of the generator model, which is inaccurate.

We propose to use the learned initializer for ABP inference on hierarchical latent variable models. Assume $p(\boldsymbol{z}) = p(\boldsymbol{z}^1, \boldsymbol{z}^2, ..., \boldsymbol{z}^L)$ to be the factorized gaussian prior, generator network is defined to be:

$$\begin{aligned} \tilde{\boldsymbol{z}}^L &= \boldsymbol{f}^L(\boldsymbol{z}^L), \\ \tilde{\boldsymbol{z}}^\ell &= \boldsymbol{f}^\ell(\tilde{\boldsymbol{z}}^{\ell+1}, \boldsymbol{z}^\ell), \quad \ell = 1, ..., L-1, \\ \boldsymbol{x} &\sim \boldsymbol{r}(\boldsymbol{x}; \boldsymbol{f}^0(\tilde{\boldsymbol{z}}^1), \end{aligned} \quad (6)$$

where $\boldsymbol{f}$ is defined as a nonlinear function (parameterized by neural network) on concatenated vectors, and $\boldsymbol{r}$ is the

METHODS	MNIST (LECUN ET AL. 1998) MSE↓	MNIST (LECUN ET AL. 1998) FID↓	SVHN (NETZER ET AL. 2011) MSE↓	SVHN (NETZER ET AL. 2011) FID↓	CELEBA (LIU ET AL. 2015) MSE↓	CELEBA (LIU ET AL. 2015) FID↓
VAE	0.0202	-	0.0192	48.47	0.0317	69.90
LVAE (L=4)	0.0185	-	0.0178	45.08	0.0286	67.60
VLAE (L=4)	0.0184	-	0.0173	45.89	0.0282	66.81
ABP(NON-PERSISTENT)	0.0192	-	0.0190	48.78	0.0289	69.31
ABP(PERSISTENT)	0.0183	-	0.0181	44.86	0.0283	51.80
IABP [E] ($L = 1$)	0.0195	-	0.0204	44.53	0.0314	49.52
IABP [E+L] ($L = 1$)	0.0180		0.0178		0.0279	
HIABP [E+L] ($L = 2$)	0.0174	-	0.0168	44.29	0.0244	47.51
HIABP [E+L] ($L = 3$)	0.0162	-	0.0145	43.91	0.0201	46.29
HIABP [E+L] ($L = 4$)	**0.0151**	-	**0.0124**	**43.22**	**0.0183**	**45.13**

Table 1: Mean Square Error (MSE) ↓ and Frechét Inception Distance (FID) ↓ on different datasets.

Figure 4: Faithful Reconstruction of SVHN Examples

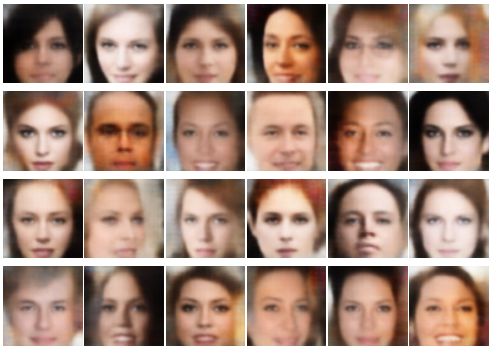


Figure 5: Unconditional Generation of HiABP [E+L] ($L = 4$) on CelebA

distribution family parameterized by $\boldsymbol{f}^0(\tilde{\boldsymbol{z}}^1)$. This design of channel-wise noise adding could be naturally used for enforcing different levels of expressiveness of neural networks. For inference network $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, we use Gaussian reparameterization in different levels:

$$\begin{aligned} \boldsymbol{h}^0 &\equiv \boldsymbol{x}, \\ \boldsymbol{h}^\ell &= \boldsymbol{g}^\ell(\boldsymbol{h}^{\ell-1}), \\ \boldsymbol{z}_0^\ell &\sim \mathcal{N}(\boldsymbol{\mu}^\ell(\boldsymbol{h}^\ell), \boldsymbol{\sigma}^\ell(\boldsymbol{h}^\ell)), \end{aligned} \tag{7}$$

where $\ell = 1, 2, ..., L$, $\boldsymbol{g}^\ell, \boldsymbol{\mu}^\ell, \boldsymbol{\sigma}^\ell$ are neural networks. Specifically, we have both the multi-layer generator model and the encoder network. For the generator model, we use the structure in Equation (6). For the encoder, we use a similar structure as in VLAE. However, unlike VLAE, which uses an encoder to directly approximate the generator posterior, we use an encoder only to provide the informative starting point for our Langevin dynamics. The difference between our model and other related models is illustrated in Figure 3.

Experiments

In this section, we apply IABP and HiABP to a series of tasks by demonstrating (1) faithful reconstruction of observed examples, (2) unconditional generation, (3) learning from incomplete data, and (4) unsupervised hierarchical latent disentanglement (conditional generation). We emphasize the simplicity of IABP and HiABP framework.

MASK TYPE	PEPPER & SALT NOISE	REGION MASK (10×10)	REGION MASK (20×20)
ERROR	0.0502	0.0487	0.0513

Table 2: Quantitative Evaluation for Learning from Incomplete Data on CelebA.

Experimental Setup

Dataset. Three datasets are employed - MNIST, Street View House Number (SVHN) (Netzer et al. 2011), and CelebA (Align & Cropped version) - with the respective training and test partitions. These datasets are expected to present increasing levels of challenge: MNIST has handwritten decimal single digits, without color, SVHN has multi-digit street numbers in several styles and colors, and CelebA has human faces in color. We code all models in Python 3.6, SciPy 1.0.0, and Tensorflow 1.15. Experiments are run on NVIDIA GTX Titan X.

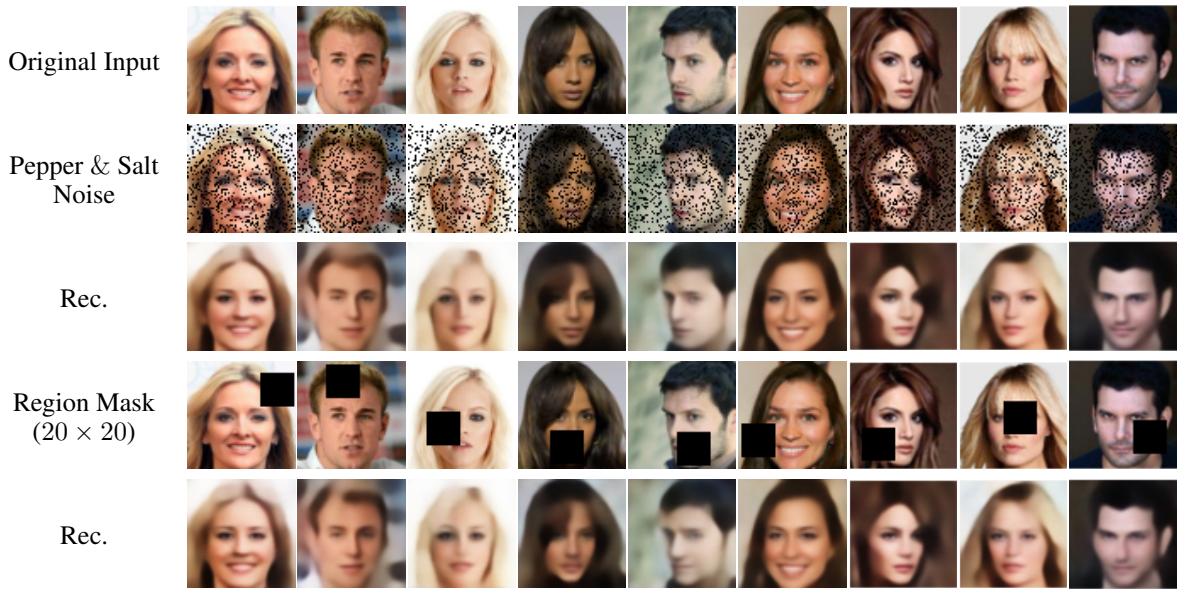


Figure 6: Recovered images on CelebA for varying occlusion masks. From top to down: (1) Pepper & Salt Noise, (2) Region Maks (20×20).

Training Details. For multi-layer HiABP, we have $\boldsymbol{z} = \{\boldsymbol{z}^\ell, \ell = 1, ..., L\}$ for which layer L is the top layer, and layer 1 is the bottom layer close to $\boldsymbol{x}$. In our case, $\boldsymbol{\mu}_\ell()$ and $\boldsymbol{\sigma}_\ell()$ are the mean vector and the diagonal variance-covariance matrix of $\boldsymbol{z}_\ell$ respectively, and $\boldsymbol{f}^\ell$ and $\boldsymbol{g}^\ell$ are deterministic layers. $\boldsymbol{f}^\ell$ is defined as two subsequent deconv2d layers with Leaky-ReLU (leaky factor 0.1) activation functions. $\boldsymbol{\mu}_l$ and $\boldsymbol{\sigma}_l$ are linear layers to project to dimensionality of $\boldsymbol{z}^l$. The final deterministic block $\boldsymbol{r}$ is a deconv2d layer with $sigmoid()$ activation function projecting to the desired dimensionality of $\boldsymbol{x}$. All (de)convolutional layers had stride = 2. IABP is a single-layer version HiABP ($L = 1$).

We train the models with 3×10^5 parameter updates optimized by Adamax (Kingma and Ba 2015). The learning rate $\eta_{\boldsymbol{\phi}} = \eta_{\boldsymbol{\theta}} = 0.0003$. If not stated otherwise, we use Langevin inference steps $T = 15$, $\sigma = 0.3$, step size $s = 0.15$, batch size $m = 100$.

Reconstruction and Unconditional Generation

We evaluate the accuracy of the learned inference dynamics $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ by reconstructing test images. In contrast to traditional MCMC posterior sampling with persistent chains, IABP with small T allows for efficient learning on the training examples. The same dynamics can be also adopted for the test examples. The increase in the number of layers contributes to the quality of reconstruction, which is quantitatively confirmed by a consistently lower Mean Square Error (MSE) in Table 1. IABP [E] and IABP [E+L] both involve encoder (E) and Langevin dynamics (L) during training, and the only difference is with or without Langevin dynamics in the test. The difference in reconstruction errors between IABP [E] and IABP [E+L] reveals the effectiveness of Langevin dynamics. Despite its simplicity, HiABP is competitive to elaborate means of inference in VAE models. Figure 4 compares the reconstructions by VAE, non-persistent ABP and HiABP [E+L] on SVHN. The fidelity of reconstructions by HiABP appears qualitatively improved over VAE.

To quantify the realism of our generated images and how well they capture the internal statistics of the training image, we evaluate the fidelity of unconditional generated examples on various datasets. To distinguish from unsupervised hierarchical latent disentanglement (*conditional generation*) in the following section, conventional generation task (randomly sampled from $\mathcal{N}(0, 1)$ without condition on layer-wise latent code) is called *unconditional generation* in this paper. Figure 5 depicts samples generated by HiABP for CelebA. Table 1 compares the Frechét Inception Distance (FID) (Heusel et al. 2017) with Inception v3 classifier on $40,000$ generated examples. Note that the only difference between IABP[E+L] and IABP[E] is whether the inference process contains Langevin dynamics or not, while there is no difference in the generation process between IABP [E] and IABP [E+L] and thus their FID results are the same. The generated images also match the true data well and visually appear better than these competing approaches.

The Impact of Langevin Steps

Langevin step number T has an important effect on the accuracy of sampling. To evaluate the impact of Langevin steps for HiABP, we take SVHN as an example and fix Langevin step number to T during training and evaluation. After training for 3×10^5 parameter updates, the model is evaluated with Langevin step number T. The influence of the Langevin step number T quantified by reconstruction Mean Square Error (MSE) on SVHN dataset is reported in Tables 3. Increasing the number of inference steps T up to 15 steps results in relative significant improvements, while $T > 15$ ap-

Figure 7: HiABP on MNIST. Each sub-figure corresponds to images generated when fixing latent code on all layers except for one. From left to right, the randomly sampled layer goes from the bottom layer to the top layer. **Right** panel: the third layer encodes digit identity; **Center** panel: the second layer encodes digit width; **Left** panel: the first (bottom) layer encodes stroke width.

LANGEVIN STEPS	$T = 5$	$T = 10$	$T = 15$	$T = 30$
MSE $\downarrow$	0.0216	0.0183	0.0168	0.0166

Table 3: The Effect of Different Langevin Steps T on HiABP-SVHN.

pears to affect the scores only marginally.

Learning from Incomplete Data

Our method can "inpaint" occluded image regions. To recover the occluded pixels, the only required modification of Equation (2) involves the computation of $||x - g_{\boldsymbol{\theta}}(z)||^2/\sigma^2$. For partially observed images, we only compute the summation over the observed pixels. We evaluate our method on $10,000$ images randomly selected from CelebA dataset. Experiments with two types of occlusions are designed: (1) Pepper $\&$ Salt occlusion, where we randomly place masks on the 64×64 image domain to cover roughly 20% of pixels. (2) Region mask occlusion, where we randomly place a 10×10 or 20×20 mask on the 64×64 image domain. Figure 6 depicts test images taken from CelebA for which a mask randomly occludes pixels in various occlusion patterns. We define "recovery error" as per-pixel difference between the original image and the recovered image on the occluded pixels. Note that the recovery errors here are not training errors, because the intensities of the occluded pixels are not observed in training. Quantitative results are reported as Table 2.

Unsupervised Hierarchical Latent Disentanglement

Convolution Neural Network (CNN) is prevailing in the past few years, and researchers put great effort to open its black box and found empirically that each layer tends to learn more abstract features. But CNN is a bottom-up process in nature which is not top-down generative. An important question is hence to directly learn hierarchical features from generative models.

Hierarchical Initialized Alternating Back-propagation (HiABP) is able to unsupervised learn highly interpretable and disentangled hierarchical features on natural image datasets with no task-specific regularization or prior knowledge, which goes far beyond the capacity of previous ABP methods. Specifically, due to the fact that different datasets have different number of semantic levels, we use $L = 3$ for HiABP-MNIST. In Figure 7, we visualize HiABP ($L = 3$) unsupervised hierarchical latent disentanglement (*conditional generation*) results on MNIST. The visualizations are generated by randomly sampling the latent code for one layer while freezing the latent code in other layers. From the visualization, we see that the three layers encode stroke width, digit width, and digit identity, respectively. These features are highly disentangled. For example, the latent code at the bottom layer controls stroke width. Modifying the code from the bottom layer while keeping the other layers fixed will generate a set of images that have different stroke width in general. Sampling latent codes at the second layer will control the digit width. Sampling latent codes at the third layer will control digit identity.

Conclusion

In this paper, we propose a novel Hierarchical Initialized Alternating Back-propagation (HiABP) framework for unsupervised representation learning. We consider an alternative to learning structured features by leveraging the expressive power of a neural network, which substantially improve the capacity of previous ABP methods. Experiments demonstrate its promising practical value on unconditional generation, reconstruction, learning from incomplete data, and unsupervised disentangled representation learning. With both high efficiency and exactness, our work paves the way for future down-stream applications.

Acknowledgments

This work is supported in part by the ECS through the Research Grants Council of Hong Kong under Grant No.24206219 and in part by InnoHK CPII Grant.

References

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Caterini, A. L.; Doucet, A.; and Sejdinovic, D. 2018. Hamiltonian Variational Auto-Encoder. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 8167–8177. Curran Associates, Inc. URL http://papers.nips.cc/paper/8039-hamiltonian-variational-auto-encoder.pdf.

Han, T.; Lu, Y.; Zhu, S.-C.; and Wu, Y. N. 2017. Alternating back-propagation for generator network. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Han, T.; Nijkamp, E.; Fang, X.; Hill, M.; Zhu, S.-C.; and Wu, Y. N. 2019. Divergence Triangle for Joint Training of Generator Model, Energy-Based Model, and Inferential Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Han, T.; Nijkamp, E.; Zhou, L.; Pang, B.; Zhu, S.-C.; and Wu, Y. N. 2020. Joint Training of Variational Auto-Encoder and Latent Energy-Based Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7978–7987.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 6626–6637.

Hinton, G.; Dayan, P.; Frey, B.; and Neal, R. 1995. The "wake-sleep" algorithm for unsupervised neural networks. *Science* 268(5214): 1158–1161. ISSN 0036-8075. doi:10.1126/science.7761831. URL https://science.sciencemag.org/content/268/5214/1158.

Hoffman, M. D. 2017. Learning Deep Latent Gaussian Models with Markov Chain Monte Carlo. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1510–1519. International Convention Centre, Sydney, Australia: PMLR.

Kim, H.; and Park, H. 2008. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM journal on matrix analysis and applications* 30(2): 713–730.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Kingma, D. P.; and Welling, M. 2014. Auto-encoding Variational Bayes. In *International Conference on Learning Representations*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11): 2278–2324.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Neal, R. 2012. MCMC Using Hamiltonian Dynamics. *Handbook of Markov Chain Monte Carlo* doi:10.1201/b10905-6.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Nijkamp, E.; Hill, M.; Zhu, S.-C.; and Wu, Y. N. 2019. Learning Non-Convergent Non-Persistent Short-Run MCMC Toward Energy-Based Model. In *Advances in Neural Information Processing Systems 32*, 5232–5242. Curran Associates, Inc.

Nijkamp, E.; Pang, B.; Han, T.; Zhu, S.-C.; and Wu, Y. N. 2020. Learning Multi-layer Latent Variable Model via Variational Optimization of Short Run MCMC for Approximate Inference. *ECCV* .

Olshausen, B. A.; and Field, D. J. 1997. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research* 37(23): 3311–3325.

Rezende, D.; and Mohamed, S. 2015. Variational Inference with Normalizing Flows. volume 37 of *Proceedings of Machine Learning Research*, 1530–1538. Lille, France: PMLR. URL http://proceedings.mlr.press/v37/rezende15.html.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In Xing, E. P.; and Jebara, T., eds., *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, 1278–1286. Bejing, China: PMLR.

Ruiz, F.; and Titsias, M. 2019. A Contrastive Divergence for Combining Variational Inference and MCMC. volume 97 of *Proceedings of Machine Learning Research*, 5537–5545. Long Beach, California, USA: PMLR. URL http://proceedings.mlr.press/v97/ruiz19a.html.

Sø nderby, C. K.; Raiko, T.; Maaløe, L.; Sø nderby, S. r. K.; and Winther, O. 2016. Ladder Variational Autoencoders. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*, 3738–3746. Curran Associates, Inc.

Xie, J.; Gao, R.; Zheng, Z.; Zhu, S.-C.; and Wu, Y. N. 2019. Learning Dynamic Generator Model by Alternating Back-Propagation Through Time. *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)* .

Xing, X.; Gao, R.; Han, T.; Zhu, S. C.; and Wu, Y. N. 2020. Deformable Generator Networks: Unsupervised Disentanglement of Appearance and Geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1–1. doi:10.1109/TPAMI.2020.3013905.

Xing, X.; Han, T.; Gao, R.; Zhu, S.-C.; and Wu, Y. N. 2019. Unsupervised disentangling of appearance and geometry by deformable generator network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 10354–10363.

Xing, X.; Wu, T.; Zhu, S.-C.; and Wu, Y. N. 2020. Inducing Hierarchical Compositional Model by Sparsifying Generator Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, S.; Song, J.; and Ermon, S. 2017. Learning hierarchical features from deep generative models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 4091–4099. JMLR. org.