Co-Generation with GANs using AIS based HMC

Tiantian Fang, Alexander G. Schwing
University of Illinois at Urbana-Champaign
{tf6, aschwing}@illinois.edu

Abstract

Inferring the most likely configuration for a subset of variables of a joint distribution given the remaining ones – which we refer to as co-generation – is an important challenge that is computationally demanding for all but the simplest settings. This task has received a considerable amount of attention, particularly for classical ways of modeling distributions like structured prediction. In contrast, almost nothing is known about this task when considering recently proposed techniques for modeling high-dimensional distributions, particularly generative adversarial nets (GANs). Therefore, in this paper, we study the occurring challenges for co-generation with GANs. To address those challenges we develop an annealed importance sampling (AIS) based Hamiltonian Monte Carlo (HMC) co-generation algorithm. The presented approach significantly outperforms classical gradient-based methods on synthetic data and on CelebA.

1 Introduction

While generative adversarial nets (GANs) [6] and variational auto-encoders (VAEs) [8] model a joint probability distribution which implicitly captures the correlations between multiple parts of the output, *e.g.*, pixels in an image, and while those methods permit easy sampling from the entire output space domain, it remains an open question how to sample from part of the domain given the remainder? We refer to this task as co-generation.

To enable co-generation for a domain unknown at training time, for GANs, optimization based algorithms have been proposed [15, 10]. Intuitively, they aim at finding that latent sample which accurately matches the observed part. However, successful training of the GAN leads to an increasingly ragged energy landscape, making the search for an appropriate latent variable via backpropagation through the generator harder and harder until it eventually fails.

To deal with this ragged energy landscape during co-generation, we develop a method using an annealed importance sampling (AIS) [11] based Hamiltonian Monte Carlo (HMC) algorithm [4, 12], which is typically used to estimate (ratios of) the partition function [14, 13]. Rather than focus on the partition function, the proposed approach leverages the benefits of AIS, *i.e.*, gradually annealing a complex probability distribution, and HMC, *i.e.*, avoiding a localized random walk.

We evaluate the proposed approach on synthetic data and imaging data (CelebA), showing compelling results via MSE and MSSIM metrics. For more details and results please see our main conference paper [5].

2 AIS based HMC for Co-Generation

In the following we first motivate the problem of co-generation before we present an overview of our proposed approach and discuss the details of the employed Hamiltonian Monte Carlo method.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

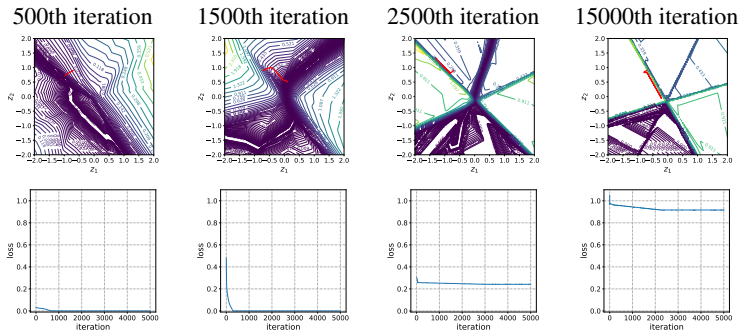


Figure 1: Vanilla GAN loss in $\mathcal{Z}$ space (top) and GD reconstruction error for generators trained for 500, 1.5k, 2.5k and 15k epochs.

2.1 Motivation

Assume we are given a well trained generator $\hat{x} = G_\theta(z)$, parameterized by θ, which is able to produce samples $\hat{x}$ from an implicitly modeled distribution $p_G(x|z)$ via a transformation of embeddings z [6, 1, 9, 2, 3]. Further assume we are given partially observed data x_o while the remaining part x_h of the data $x = (x_o, x_h)$ is latent.

To reconstruct the latent parts of the data x_h from available observations x_o, a program can be formulated as follows:

$$z^* = \arg\min_z \|x_o - G_\theta(z)_o\|_2^2, \tag{1}$$

where $G_\theta(z)_o$ denotes the restriction of the generated sample $G_\theta(z)$ to the observed part. Upon solving the program given in Eq. (1), we obtain an estimate for the missing data $\hat{x}_h = G(z^*)_h$.

However, in practice, Eq. (1) turns out to be extremely hard to address, particularly if the generator $G_\theta(z)$ is very well trained. To see this, consider as an example a generator operating on a 2-dimensional latent space $z = (z_1, z_2)$ and 2-dimensional data $x = (x_1, x_2)$ (blue points in Fig. 2(a)). We use $h = 1$ and let $x_o = x_2 = 0$. In the first row of Fig. 1 we illustrate the loss surface of the objective given in Eq. (1) obtained when using a generator $G_\theta(z)$ trained on the original 2-dimensional data for 500, 1.5k, 2.5k and 15k iterations (columns in Fig. 1).

We observe the latent space to become increasingly ragged, exhibiting folds that clearly separate different data regimes. First (*e.g.*, gradient descent (GD)) or second order optimization techniques cannot cope easily with such a loss landscape and likely get trapped in local optima. We observe GD (red trajectory in Fig. 1 first row and loss in second row) to get stuck in a local optimum as the loss fails to decrease to zero once the generator better captures the data.

To prevent those local-optima issues for co-generation, we propose an annealed importance-sampling (AIS) based Hamiltonian Monte Carlo (HMC) method in the following (Alg. 1).

2.2 Overview

In order to reconstruct the hidden portion x_h of the data $x = (x_o, x_h)$ we are interested in drawing samples $\hat{z}$ such that $\hat{x}_o = G_\theta(\hat{z})_o$ has a high probability under $\log p(z|x_o) \propto -\|x_o - G_\theta(z)_o\|_2^2$.

To obtain samples $\hat{z}$ following the posterior distribution $p(z|x_o)$, we use annealed importance sampling (AIS) [11] to gradually approach the complex and often high-dimensional posterior distribution $p(z|x_o)$ by simulating a Markov Chain starting from the prior distribution $p(z) = \mathcal{N}(z|0, I)$, a standard normal distribution (zero mean and unit variance). Formally, we define an annealing schedule for the parameter β_t from $\beta_0 = 0$ to $\beta_T = 1$. At every time step $t \in \{1, \ldots, T\}$ we refine the samples drawn at the previous timestep $t-1$ so as to represent the distribution $\hat{p}_t(z|x_o) = p(z|x_o)^{\beta_t} p(z)^{1-\beta_t}$. We use a sigmoid schedule for the parameter β_t.

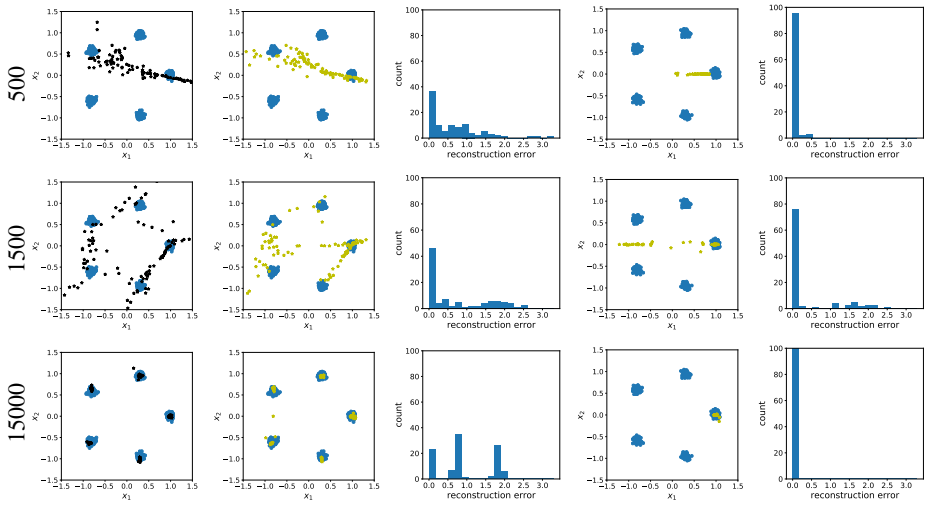
To successively refine the samples we use Hamilton Monte Carlo (HMC) sampling because a proposed update can be far from the current sample while still having a high acceptance probability. We use 0.01 as the leapfrog step size and employ 10 leapfrog updates per HMC loop for the synthetic 2D dataset and 20 leapfrog updates for real dataset at first. The acceptance rate is 0.65, as recommended by Neal [12].

2.3 Hamilton Monte Carlo

Hamilton Monte Carlo (HMC) [4] is capable of traversing folds in an energy landscape. For this, HMC methods trade potential energy $U_t(z) = -\log \hat{p}_t(z|x_o)$ with kinetic energy $K_t(v)$.

Algorithm 1 AIS based HMC

1: **Input:** $p(z|x_o)$, β_t $\forall t \in \{1, \ldots, T\}$
2: Draw set of samples $z \in \mathcal{Z}$ from prior distribution $p(z)$
3: **for** $t = 1, \ldots, T$ **do** // AIS loop
4: Define $\hat{p}_t(z|x_o) = p(z|x_o)^{\beta_t} p(z)^{1-\beta_t}$
5: **for** $m = 1, \ldots, M$ **do** // HMC loop
6: $\forall z \in \mathcal{Z}$ initialize Hamiltonian and momentum variables $v \sim \mathcal{N}(0, I)$
7: $\forall z \in \mathcal{Z}$ compute new proposal sample using leapfrog integration on Hamiltonian
8: $\forall z \in \mathcal{Z}$ use Metropolis Hastings to check whether to accept the proposal and update $\mathcal{Z}$
9: **end for**
10: **end for**
11: **Return:** $\mathcal{Z}$



(a) (b) (c) (d) (e)

Figure 2: Rows correspond to generators trained for a different number of epochs as indicated (left). The columns illustrate: (a) Samples generated with a vanilla GAN (black); (b) GD reconstructions from 100 random initializations; (c) Reconstruction error bar plot for the result in column (b); (d) Reconstructions recovered with Alg. 1; (e) Reconstruction error bar plot for the results in column (d).

HMC defines a Hamiltonian $H(z, v) = U(z) + K(v)$ or conversely a joint probability distribution $\log p(z, v) \propto -H(z, v)$ and proceeds by iterating three steps M times.

In a first step, the Hamiltonian is initialized by randomly sampling the momentum variable v, typically using a standard Gaussian. In the second step, (z^*, v^*) are proposed via leapfrog integration to move along a hypersurface of the Hamiltonian. In the final third step we decide whether to accept the proposal (z^*, v^*) computed via leapfrog integration. Formally, we accept the proposal with probability

$$\min\{1, \exp\left(-H(z^*, v^*) + H(z, v)\right)\}. \tag{2}$$

If the proposed state (z^*, v^*) is rejected, the $m+1$-th iteration reuses z, otherwise z is replaced with z^* in the $m+1$-th iteration. This process is shown in Alg. 1 line 6, 7 and 8.

3 Experiments

Baselines: In the following, we evaluate the proposed approach on synthetic and imaging data. We use two GD baselines, employing different initialization methods. The first one samples a single z randomly. The second picks that one sample z from 5000 points which best matches the objective given in Eq. (1) initially.

3.1 Synthetic Data

To illustrate the advantage of our proposed method over the common baseline, we first demonstrate our results on 2-dimensional synthetic data. Specifically, the 2-dimensional data $x = (x_1, x_2)$ is drawn from a mixture of five equally weighted Gaussians each with a variance of 0.02, the means of

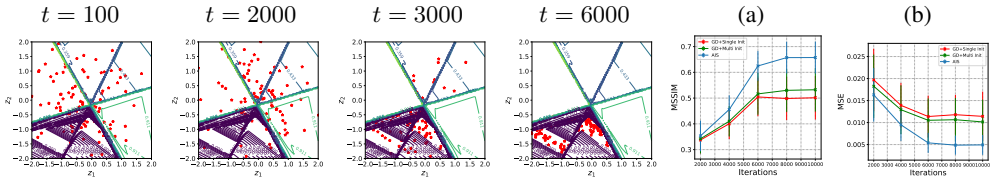


Figure 3: Samples z in $\mathcal{Z}$ space during the AIS procedure: after 100, 2k, 3k, and 6k AIS loops. (a) MSSIM reconstructions errors on CelebA (b) MSE reconstructions errors over Progressive GAN training iterations.

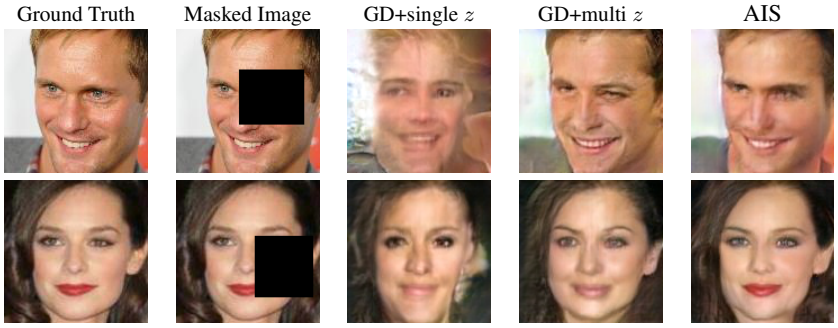


Figure 4: Reconstructions on 128×128 CelebA images for a progressive GAN trained for 10k iterations.

which are spaced equally on the unit circle. See the blue points in columns (a), (b), and (d) of Fig. 2 for an illustration.

In this experiment, we aim to reconstruct $x = (x_1, x_2)$, given $x_o = x_2 = 0$. The optimal solution for the reconstruction is $\hat{x} = (1, 0)$, where the reconstruction error should be 0. However, as discussed in reference to Fig. 1 earlier, we observe that energy barriers in the $\mathcal{Z}$-space complicate optimization.

In contrast, our proposed AIS co-generation method only requires one initialization to achieve the desired result after $6,000$ AIS loops, as shown in Fig. 2 (15000 (d)). Specifically, reconstruction with generators trained for a different number of epochs (500, 1.5k and 15k) are shown in the rows. The samples obtained from the generator for the data (blue points in column (a)) are illustrated in column (a) using black color. Using the respective generator to solve the program given in Eq. (1) via GD yields results highlighted with yellow color in column (b). The empirical reconstruction error frequency for this baseline is given in column (c). The results and the reconstruction error frequency obtained with Alg. 1 are shown in columns (d, e). We observe significantly better results and robustness to initialization.

In Fig. 3 we show for 100 samples that Alg. 1 moves them across the energy barriers during the annealing procedure, illustrating the benefits of AIS based HMC over GD.

3.2 Imaging Data

To validate our method on real data, we evaluate on CelebA, using MSE and MSSIM metrics. We use the progressive GAN architecture [7]. The size of the input is 512 and the size of the output is 128 × 128. We randomly mask blocks of width and height ranging from 30 to 60. Then we use Alg. 1 for reconstruction with 500 HMC loops.

In Fig. 3 (a,b), we observe that Alg. 1 outperforms over both baselines for all GAN training iterations on both MSSIM and MSE metrics. In Fig. 4 we show results generated by both baselines and Alg. 1.

4 Conclusion

We propose a co-generation approach, *i.e.*, we complete partially given input data, using annealed importance sampling (AIS) based on the Hamiltonian Monte Carlo (HMC). Different from classical optimization based methods, specifically GD, which get easily trapped in local optima when solving this task, the proposed approach is much more robust. Importantly, the method is able to traverse large energy barriers that occur when training generative adversarial nets. Its robustness is due to AIS gradually annealing a probability distribution and HMC avoiding localized walks.

Acknowledgments: This work is supported in part by NSF under Grant No. 1718221 and MRI #1725729, UIUC, Samsung, 3M, Cisco Systems Inc. (Gift Award CG 1377144) and Adobe. We thank NVIDIA for providing GPUs used for this work and Cisco for access to the Arcetri cluster.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. In *ICML*, 2017.

[2] I. Deshpande, Z. Zhang, and A. G. Schwing. Generative Modeling using the Sliced Wasserstein Distance. In *Proc. CVPR*, 2018.

[3] I. Deshpande, Y.-T. Hu, R. Sun, A. Pyrros, N. Siddiqui, S. Koyejo, Z. Zhao, D. Forsyth, and A. G. Schwing. Max-Sliced Wasserstein Distance and its use for GANs. In *Proc. CVPR*, 2019.

[4] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 1987.

[5] T. Fang and A. G. Schwing. Co-generation with gans using ais based hmc. In *NeurIPS*, 2019.

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. In *https://arxiv.org/abs/1406.2661*, 2014.

[7] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2017.

[8] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *https://arxiv.org/abs/1312.6114*, 2013.

[9] Y. Li, A. G. Schwing, K.-C. Wang, and R. Zemel. Dualing GANs. In *Proc. NeurIPS*, 2017.

[10] M.-Y. Liu and O. Tuzel. Coupled Generative Adversarial Networks. In *Proc. NIPS*, 2016.

[11] R. M. Neal. Annealed Importance Sampling. *Statistics and Computing*, 2001.

[12] R. M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2010.

[13] J. Sohl-Dickstein and B. Culpepper. Hamiltonian annealed importance sampling for partition function estimation. In *https://arxiv.org/abs/1205.1925*, 2012.

[14] Y. Wu, Y. Burda, R. Salakhutdinov, and R. B. Grosse. On the quantitative analysis of decoder-based generative models. In *ICLR*, 2017.

[15] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic Image Inpainting with Deep Generative Models. In *Proc. CVPR*, 2017.

5 Appendix: Additional Real data examples

We show additional results for real data experiments. We observe our proposed algorithm to recover masked images more accurately than baselines and to generate better high-resolution images given low-resolution images.

We show masked CelebA (Fig. 5) and LSUN (Fig. 6) recovery results for baselines and our method, given a Progressive GAN generator. Note that our algorithm is pretty robust to the position of the z initialization, since the generated results are consistent in Fig. 5.

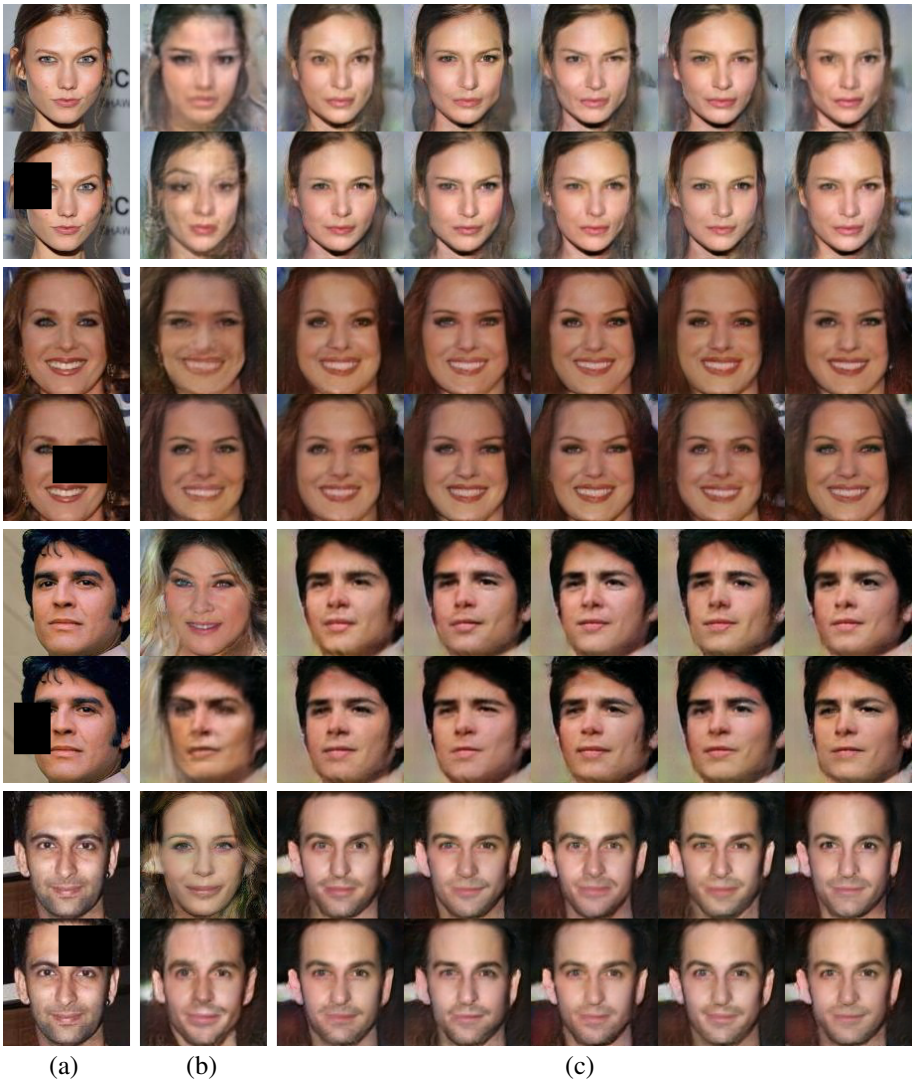


Figure 5: Reconstructions on 128×128 CelebA images for a trained PGAN at 10k-th iteration. (a) Ground truth and masked (observed) images (top to bottom); (b) The result obtained by optimizing the best z picked from 5,000 initializations (top to bottom) (c) Results generated by our algorithm.

(a) (b) (c) (d) (e)

Figure 6: Reconstructions on 256×256 LSUN images for a trained PGAN at 10k-th iteration. (a) ground truth; (b) masked (observed) images; (c) the result optimized by single z; (d) the result obtained by optimizing the best z picked from 5,000 initializations; (e) result of our algorithm.