Mask CycleGAN : Unpaired Multi-modal Domain Translation with Interpretable Latent Variable

Minfa Wang
Stanford University
minfa@stanford.edu

1 Introduction

CycleGAN [4] is a popular approach for unpaired image-to-image translation between two domains. It has been proven to be effective in a wide variety of domain translation tasks, including horse-zebra, apple-orange, summer-winter, etc. While it keeps inspiring generative modelling community to build up more and more applications and research ideas, CycleGAN has its limitations too. One notable limitation is that the translation is deterministic and hence lack of variation. People have discovered that achieving multi-modality through CycleGAN is challenging [1], largely due to the reason that the source image is at high dimensionality and usually causes the generator network to ignore other noise sampled from other distribution.

There are several attempts in the community to address this limitation, some of which are detailed in the Related Work section below. Then we will propose our idea, **Mask CycleGAN**, which enables unpaired multi-modal domain translation with the unique property that the latent variable is interpretable.

2 Related Work

CycleGAN [4] is a popular approach for unpaired image-to-image translation between two domains. One of the most innovative ideas from this work is *cycle consistency*, which encourages the mapping between two domains to be invertible, and indirectly alleviate the problem of mode collapse. CycleGAN is capable of generating visually appealing images.

Augmented CycleGAN [1] brings multi-modality into CycleGAN by augmenting it with two latent variables Z_a and Z_b, and corresponding encoders E_a, E_b and discriminators D_{Z_a}, D_{Z_b}. The latent variables are optimized through the minimax game between the encoder and the discriminator.

BicycleGAN [5] is an architecture for paired multi-modal image-to-image translation. The name Bicycle refers to the two cycles from 1) Conditional Variational Autoencoder GAN (*cVAE-GAN*) and 2) Conditional Latent Regressor GAN (*cLR-GAN*).

Image inpainting is a technique to reconstruct a image patch from a partially covered or blurred image. The input of image inpainting can be thought as a clear source image applied on a mask, which is similar to our setup. Some of the works [3] in this field inspires the generator design of the Mask CycleGAN to impose soft constraint on pixel invariance on certain image region during transformation.

3 Problem statement

A natural idea to enable multi-modality in image generation task is to introduce some additional latent variable. A popular choice [1, 5] for the distribution of the latent variable is Gaussian distribution,

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

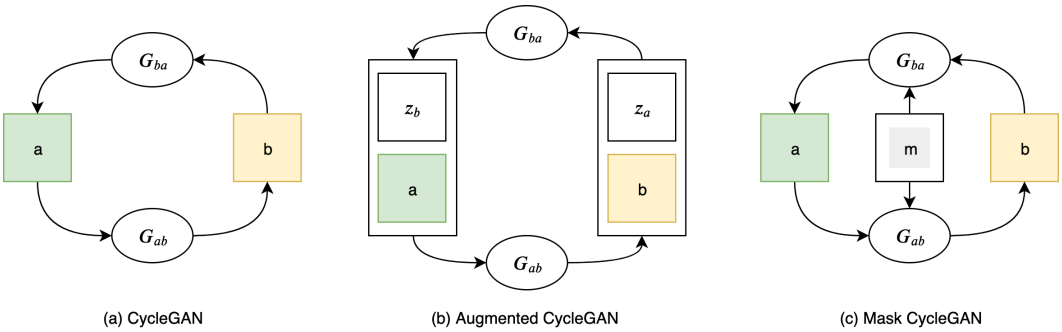

Figure 1: Comparison of variants of CycleGAN.

because it is easy to sample and evaluate. The downsides are 1) the sample is not interpretable and 2) the effect of the latent variable is hard to control. In this work, we aim to use mask as the latent variable to combat the interpretability issue. Please see Figure 1 for a comparison of high-level architecture of different variants of CycleGAN including ours.

There are two remarks about **Mask CycleGAN**. First, because the mask is interpretable, we do not need to make assumptions about its ground truth distribution family, but could generate meaningful samples directly. Second, the architecture is designed to be a full generalization of CycleGAN and could be easily shown that with certain choices of parameters, our method could gracefully fall back to CycleGAN, making the model at least as expressive as original CycleGAN.

To make the scope of work tractable, we limit the mask by having certain shape, size and position. We use popular domain translation datasets like MNIST-SVHN and horse-zebra as our datasets. We evaluate the model both qualitatively (by judging the realism and aesthetics of the generated images) and quantitatively (by evaluating on FCN [2] score). We expect to see that with different choices of masks, the generator produces outputs that is mostly conditioned on the masked region of the source image and invariant of the context (non-masked) region of the source image.

4 Approach

We introduce the following terminologies to help elaborate the technical approach. Same as CycleGAN, we have a as a sample image from the source domain, b as a sample from the destination domain, and G_{AB} as generator mapping an image from source to destination. With mask m, we could derive the following properties: $\tilde{b} = G_{AB}(a, m)$ as the fake image in destination domain, $a' = G_{BA}(\tilde{b}, m)$ as the recovered image in source domain. Other terms like $\tilde{a}$, b' and G_{BA} are defined symmetrically.

4.1 Generator

The generator design is one of the critical parts of the architecture. Figure 2(a) shows the plain-vanilla generator used in original CycleGAN that maps a source image to a destination image. It is easy to see that the mapping is deterministic with a fixed source image. In literature, when people try to introduce the additional latent variable z, a common approach [5, 1] is concatenates the latent variable vector to some intermediate vector representation of the source image. The drawbacks of this approach are 1) the influence of latent variable is hard to interpret and control, and 2) empirically the generator often tends to ignore variations in the latent variable.

Since our latent variable, m, is interpretable, we could design its interaction with the source image in a way that enforces the masking behavior. The interaction is defined through the **mask encoder** E, with its architecture detailed in Figure 2(b). The full generator design is shown in Figure 2(c), where complicated domain translation logic, captured in G, is required to only depend on masked region of the source image.

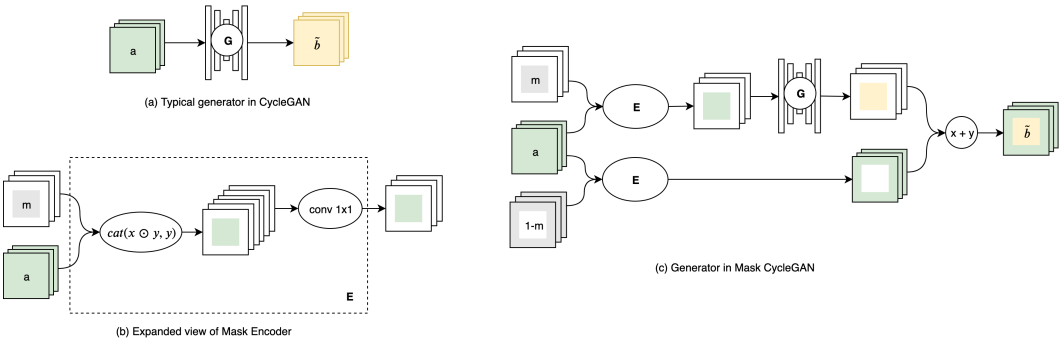

Figure 2: Comparison of generators. (b) is the expanded view of the mask encoder E used in (c).

4.2 Losses

Similar to CycleGAN, we optimize the model by minimizing three pairs of losses. We have made modifications on each pair to accommodate the architecture change introduced by the mask.

GAN loss For each triplet $(a, \tilde{a}, m)$, we will have two discriminators D_{AM} and D_{AF}. The first generator D_{AF} is functionally equivalent to the discriminator D_A in original CycleGAN, which is trying to distinguish a true image from a generated one for the source distribution, and implicitly encourages the generator to produce overall coherent image inside and outside the masked area. The second discriminator D_{AM} is responsible for the same task for the masked image pair. We have the final GAN loss as the normalized weighted sum of the two discriminators' losses. Formally:

$$\mathcal{L}_{GAN}^{AF} = -\mathbb{E}_{a\sim A}[\log D_{AF}(a)] - \mathbb{E}_{\tilde{a}\sim\tilde{A}}[\log(1 - D_{AF}(\tilde{a}))]$$
$$\mathcal{L}_{GAN}^{AM} = -\mathbb{E}_{a\sim A}[\log D_{AM}(a \odot m)] - \mathbb{E}_{\tilde{a}\sim\tilde{A}}[\log(1 - D_{AM}(\tilde{a} \odot m))]$$
$$\mathcal{L}_{GAN}^{A} = \lambda_{GAN}^{M}\mathcal{L}_{GAN}^{AM} + (1 - \lambda_{GAN}^{M})\mathcal{L}_{GAN}^{AF}$$

There are two remarks. First, we need to introduce the second discriminator D_{AM} because despite the fact that generator tries to produce coherent images, there would generally exists some discontinuity around the mask boundary, making the task relatively easy for D_A, and in turn causes gradient vanishing problems to the generator. Even if the generator manages to learn, without D_{AM}, the generator would tend to ignore the mask and suffer from mode collapse. Second, when $\lambda_{GAN}^{M}L_{GAN}^{AM} = 0$, this objective falls back to the GAN objective in original CycleGAN.

Cycle loss facilitates cycle-consistency for translation between two domains. The high-level idea is that when a image goes through a forward mapping and then a backward mapping, it should recover to itself, which requires the forward mapping to preserve the information of the source image. This is an effective method to prevent mode collapse. Our loss objective is the normalized weighted sum of the cycle losses for the masked and context areas.

$$\mathcal{L}_{CYC}^{A} = \lambda_{CYC}^{M}||(a - a') \odot m||_1 + (1 - \lambda_{CYC}^{M})||(a - a') \odot (1 - m)||_1$$

Different weights are applied for masked and context areas, and the weights could be adjusted to control how strict we hope the generator to keep the pixels in the context area intact. When $\lambda_{CYC}^{M} = 0.5$, this loss falls back to the cycle loss in original CycleGAN (with a scale factor).

Identity loss states that when you try to map an image from the destination domain to the destination domain, it should map to itself. Formally:

$$\mathcal{L}_{IDT}^{A} = ||a - G_{BA}(a, m)||_1$$

Note this formulation is identical to the identity loss in original CycleGAN. We don't need special treatment for the masked area because we intentionally want this specific mapping to be invariant of the mask.

4.3 Full objective

The full objective is

$$\mathcal{L} = (\mathcal{L}^A_{GAN} + \mathcal{L}^B_{GAN}) + \lambda_{CYC}(\mathcal{L}^A_{CYC} + \mathcal{L}^B_{CYC}) + \lambda_{IDT}(\mathcal{L}^A_{IDT} + \mathcal{L}^B_{IDT})$$

In our experiments, we used $\lambda^M_{GAN} = 0.7$, $\lambda^M_{CYC} = 0.3$, $\lambda_{CYC} = 10$ and $\lambda_{IDT} = 5$.

4.4 Algorithm

Algorithm 1: Masked CycleGAN

initialize weights and optimizers ;
initialize G_{AB}, G_{BA}, D_{AF}, D_{BF}, D_{AM}, D_{BM} ;
while *True* **do**
  Read next batch a, b ;
  Sample m from M ;
  $\tilde{b}$ = $G_{AB}(a, m)$, $\tilde{a}$ = $G_{BA}(b, m)$;
  a' = $G_{BA}(\tilde{b}, m)$, b' = $G_{AB}(\tilde{a}, m)$;
  `// -- optimize generator below --`
  l_cyc_a = λ^M_{CYC} * cyc_loss(a * m, a' * m) + (1 - λ^M_{CYC}) * cyc_loss(a * (1 - m), a' * (1 - m)) ;
  l_cyc_b = λ^M_{CYC} * cyc_loss(b * m, b' * m) + (1 - λ^M_{CYC}) * cyc_loss(b * (1 - m), b' * (1 - m)) ;
  l_idt = idt_loss(a, G_{BA}(a, m)) + idt_loss(b, G_{AB}(b, m)) ;
  l_gan_a = (1 - λ^M_{GAN}) * gan_loss($D_{AF}(\tilde{a})$) + λ^M_{GAN} * gan_loss($D_{AM}(\tilde{a}$ * m)) ;
  l_gan_b = (1 - λ^M_{GAN}) * gan_loss($D_{BF}(\tilde{b})$) + λ^M_{GAN} * gan_loss($D_{BM}(\tilde{b}$ * m)) ;
  l_g = (l_gan_a + l_gan_b) + λ_{CYC} * (l_cyc_a + l_cyc_b) + λ_{IDT} * l_idt ;
  l_g.backward() ;
  optimizer_g.step(l_g) ;
  `// -- optimize discriminator below --`
  l_da = (1 - λ^M_{GAN}) * d_loss(D_{AF}, a, $\tilde{a}$) + λ^M_{GAN} * d_loss(D_{AM}, a * m, $\tilde{a}$ * m) ;
  l_db = (1 - λ^M_{GAN}) * d_loss(D_{BF}, b, $\tilde{b}$) + λ^M_{GAN} * d_loss(D_{BM}, b * m, $\tilde{b}$ * m) ;
  l_d = l_da + l_db ;
  l_d.backward() ;
  optimizer_d.step(l_d) ;
end

5 Preliminary results

Technically, the only restriction necessary for the mask m is that it needs to have the same shape as the source image. In its most general form, the mask could have irregular shape, and could scatter around several discontinued regions on the image. However, for the purpose of making this experiment more tractable, at the current stage we restrict the mask to be always square, and is centered in the image. We obtained qualitative results on MNIST-SVHN and Horse-Zebra datasets. We plan to obtain additional quantitative results in the next a few weeks.

5.1 MNIST-SVHN

We chose image size to be 32 x 32 for this dataset. Example outputs are shown in Figure 3(a). Empirically, the generator performs better on the forward mapping (from MNSIT to SVHN), but struggles more on the backward mapping. One interesting example is that when we map a digit 2 from SVHN to MNIST with a small mask, it converts to a digit 1. One explanation is the masked region of digit 2 in SVHN is a blurb of white pixels, the generator chose to generate digit 1 rather than a blurb in the hope to fool the discriminator.

5.2 Horse-Zebra

We chose the image size to be 128 X 128 for this dataset. Example outputs are shown in Figure 3(b). The generator generally performs better on the forward mapping (from horse to zebra). This seems

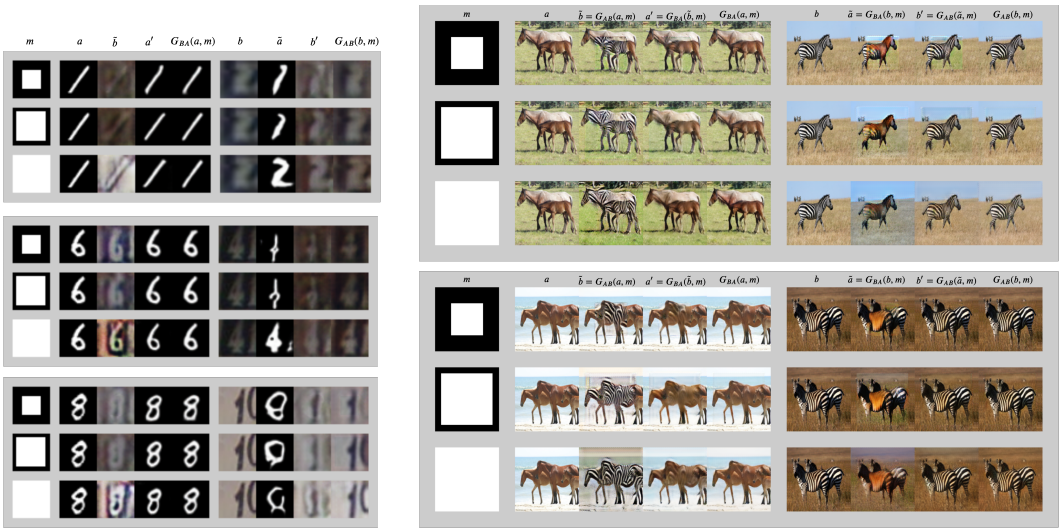

Figure 3: Outputs of Mask CycleGAN. (a) shows the results from MNIST-SVHN translation. (b) shows horse-zebra translation. Please zoom in to inspect the details.

to be due to the fact that most zebra pictures in the training dataset have the clear white and black strips pattern, whereas the patterns for a horse image is more diverse and hard to extract. However, the model still shows some promising results in many cases. When the mask is small, the generator produces fascinating pictures of creatures that is half-horse and half-zebra.

References

[1] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. C. Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. *CoRR*, abs/1802.10151, 2018. URL `http://arxiv.org/abs/1802.10151`.

[2] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014. URL `http://arxiv.org/abs/1411.4038`.

[3] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. *CoRR*, abs/1604.07379, 2016. URL `http://arxiv.org/abs/1604.07379`.

[4] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017. URL `http://arxiv.org/abs/1703.10593`.

[5] J. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. *CoRR*, abs/1711.11586, 2017. URL `http://arxiv.org/abs/1711.11586`.