Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation

Jianwen Xie [1*], **Zilong Zheng** [2*], **Xiaolin Fang** [3], **Song-Chun Zhu** [2,4,5], **Ying Nian Wu** [2]
[1] Cognitive Computing Lab, Baidu Research, Bellevue, USA
[2] University of California, Los Angeles, USA
[3] Massachusetts Institute of Technology, Cambridge, USA
[4] Tsinghua University, Beijing, China
[5] Peking University, Beijing, China
jianwen@ucla.edu, z.zheng@ucla.edu, xiaolinf@csail.mit.edu, sczhu@stat.ucla.edu, ywu@stat.ucla.edu

Abstract

This paper studies the unsupervised cross-domain translation problem by proposing a generative framework, in which the probability distribution of each domain is represented by a generative cooperative network that consists of an energy-based model and a latent variable model. The use of generative cooperative network enables maximum likelihood learning of the domain model by MCMC teaching, where the energy-based model seeks to fit the data distribution of domain and distills its knowledge to the latent variable model via MCMC. Specifically, in the MCMC teaching process, the latent variable model parameterized by an encoder-decoder maps examples from the source domain to the target domain, while the energy-based model further refines the mapped results by Langevin revision such that the revised results match to the examples in the target domain in terms of the statistical properties, which are defined by the learned energy function. For the purpose of building up a correspondence between two unpaired domains, the proposed framework simultaneously learns a pair of cooperative networks with cycle consistency, accounting for a two-way translation between two domains, by alternating MCMC teaching. Experiments show that the proposed framework is useful for unsupervised image-to-image translation and unpaired image sequence translation.

1 Introduction

Cross-domain translation, such as image-to-image translation, has shown its importance over the last few years on numerous computer vision and computer graphics tasks which require translating an example from one domain to another, for example, neural style transfer, photo enhancing, etc. This problem can be solved by learning a conditional generative model as a mapping from source domain to target domain in a supervised manner, when paired training examples between two domains are available. However, manually pairing up examples between two domains is costly in both time and efforts, and in some cases it is even impossible. For example, learning to translate a photo to a Van Gogh style painting requires plenty of real scene photos paired with their corresponding paintings for training. Therefore, unsupervised cross-domain translation is considered more applicable since different domains of independent data collections are easily accessible, yet it is also regarded as a harder problem due to the lack of supervision on instance-level correspondence between different domains. This paper focuses on unsupervised cross-domain translation problem where paired training examples are not available.

With the recent success of Generative Adversarial Networks (GANs) (**??**) in image generation (**???**), researchers have proposed unsupervised cross-domain translation networks based on GANs and obtained compelling results (**???**). For the sake of learning probability distribution, instead of maximizing the data likelihood, GANs introduce the concept of adversarial learning between a generator and a discriminator. Specifically, the generator is the desired implicit data distribution that maps the Gaussian prior on a low-dimensional latent space to the data space via a non-linear transformation, while the discriminator aims at distinguishing the real examples and the "fake" examples synthesized by the generator. The generator gets improved in terms of the capacity of data generation, by learning to deceive the discriminator which also evolves against the generator in such an adversarial learning scheme.

Recently, learning energy-based models (EBMs), with energy functions parameterized by modern convolutional neural networks, for explicit data probability distributions has received significant attention in the fields of computer vision and machine learning. **??????** suggest that highly realistic examples can be generated by Markov chain Monte Carlo (MCMC) sampling from the learned EBMs. **?** propose the Cooperative Networks (CoopNets) framework to learn the EBM simultaneously with a generator model in a cooperative learning scheme, where the generator plays the role of a fast sampler to initialize the MCMC sampling of the EBM, while the EBM teaches the generator via a finite-step MCMC. Within this cooperative learning process, the EBM learns from the training data, while the generator learns from the MCMC sampling of the EBM. In other words, the EBM distills the MCMC into the generator, such that the generator becomes the amortized sampler of the EBM.

Compared to adversarial learning, the energy-based co-

[*] Equal contributions

operative learning has numerous conceptual advantages for modeling and learning data distribution: (1) *Free of mode collapse*. The training of GANs is known to be difficult, unstable and easy to encounter mode collapse issue (**?**). Different from GANs, both EBM and generator in the cooperative learning framework are trained generatively by maximum likelihood. Thus, the CoopNets framework is stable and does not suffer from mode collapse problem (**?**). (2) *MCMC refinement*. Even though both GAN and CoopNets consist of two sub-models, their interactions in these two frameworks are essentially different. GAN will discard the discriminator once the generator is well-trained, while for the CoopNets, no matter whether at the training stage or testing stage, the EBM enables a refinement for the generator by the iterative MCMC sampling. (3) *Fast-thinking initializer and slow-thinking solver*. Solving a challenging problem usually requires an iterative algorithm. This amounts to slow thinking. However, the iterative algorithm usually needs a good initialization for quick convergence. The initialization amounts to fast thinking. Thus integrating fast-thinking initialization and slow-thinking sampling or optimization is very compelling. **?** point out that the cooperative learning framework corresponds to a fast-thinking and slow-thinking system, where the generator serves as a fast-thinking initializer and the EBM serves as a slow-thinking solver. The problem we solve in our paper is cross-domain visual translation.

Our framework for unsupervised cross-domain translation is based on the cooperative learning scheme. We first propose to represent a one-way translator by a cooperative network that includes an energy-based model and a latent variable model, where both of them are trained via MCMC teaching. Specifically, the latent variable model, serving as a translator, maps examples from one domain to another, while the EBM, serving as a teacher, refines the mapped results by MCMC revision, so that the revised results can match to the examples of the target domain in terms of some statistical properties. By simultaneously learning a pair of cooperative networks, each of which is obligated to represent one direction of mapping between two domains, through alternating MCMC teaching, we can achieve a novel framework for unsupervised cross-domain translation. To enforce these two mapping functions to be inverse to each other, we add a cycle consistency loss (**?**) as a constraint to regularize the training of both cooperative networks. This leads to the model we call Cycle-Consistent Cooperative Networks (CycleCoopNets).

Concretely, the contributions of our paper are four-folds:

1. We present a novel energy-based generative framework, CycleCoopNets, to study unsupervised cross-domain translation problem, where we propose to represent a two-way mapping between two domains by a pair of cooperative networks with cycle consistency property, and learn them by the alternating MCMC teaching algorithm.
2. We apply our framework to a wide range of applications of unsupervised cross-domain translation, including object transfiguration, season transfer, and art style transfer.
3. We show that our model can achieve competitive quantitative and qualitative results, compared with GAN-based and flow-based (**?**) frameworks.
4. We generalize our framework to the task of unsupervised image sequence translation by combining both spatial and temporal information along with MCMC teaching for appearance translation and motion style preservation.

2 Related Work

Our work is related to the following themes of research.

GAN-based cross-domain translation. Generative Adversarial Networks (GANs) have been successfully applied to a wide range of synthesis problems in the field of computer vision. Three closely related works are Pix2Pix (**?**) CycleGAN (**?**) and RecycleGAN (**?**). By generalizing the original unconditioned GANs to the conditioned scenarios, the Pix2Pix is a framework for supervised conditional learning, which has achieved impressive results on paired image-to-image translation tasks, such as image colorization, sketch-to-photo synthesis, etc. CycleGAN is proposed to learn a two-way translator between two domains in the absence of paired examples, by jointly training two GANs, each of which accounts for one-way translation, and enforcing cycle consistency between them. Inspired by CycleGAN, RecycleGAN (**?**) is designed to learn a two-way translator between two domains of image sequences without paired examples by adding extra temporal predictive models and enforcing spatiotemporal consistency. Note that our work does not belong to the theme of adversarial learning, even though it also includes encoder-decoder structures like CycleGAN.

Energy-based synthesis. **?** propose to adopt a modern convolutional neural network to parameterize the energy function of the energy-based model and learn the model by MCMC-based maximum likelihood estimation. The resulting model is called the generative ConvNet. Compelling results have been achieved by learning the models on images (**???**), videos (**??**), 3D voxels (**??**) and point clouds (**?**). **?** learn the model with multigid sampling and **?** learn the model with short-run MCMC. Our paper is related to energy-based synthesis, because we train energy-based models as teachers for MCMC teaching in our framework.

Cooperative learning. **??** proposes the generative cooperative network (CoopNets) that trains an EBM, such as the generative ConvNet model, with the help of a generator network serving as an amortized sampler. The energy-based generative ConvNet distills its knowledge to the generator via MCMC, and this is called MCMC teaching. Recently, **?** propose a variant of CoopNets, where a variational auto-encoder (VAE) (**?**) and an EBM is cooperatively trained via MCMC teaching. **?** further propose the conditional version of CoopNets model for supervised image-to-image translation. Unlike the above approaches, our framework simultaneously trains two cooperative networks, each of which accounts for one direction of mapping between two domains, and enforces cycle consistency between the two mappings for unsupervised image-to-image translation.

Style transfer using neural networks. **?** first propose to use a ConvNet structure, which is pre-trained for image classification, to transfer the artistic style of a style image to a content image. Such a neural style transfer is achieved by synthesizing an image that matches the style of the style image and the content of the content image in terms of Gram matrix statistics of the pre-trained VGG (**?**) features. Other works include **????**. Our paper studies learning a bidirectional mapping between two domains, rather than a unidirectional mapping between two specific instances. Also, our model can be applied to not only style transfer but also other image-to-image translation tasks, e.g., object transfiguration, etc.

3 Proposed Framework

3.1 Problem Definition

Suppose we have two different domains, say $\mathcal{X}$ and $\mathcal{Y}$, and two data collections from these domains $\{x_i, i = 1, ..., n_x\}$ and $\{y_i, i = 1, ..., n_y\}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. n_x and n_y are numbers of examples in the collections, respectively. n_x and n_y are not necessarily the same. Let $p_{\text{data}}(x)$ and $p_{\text{data}}(y)$ be the unknown data distributions of these two domains. Without instance-level correspondence between two collections, we want to learn mapping functions between two domains for the sake of cross-domain translation.

3.2 Latent Variable Model as a Translator

Let us talk about one-way translation problem first. To transfer image across domains, say $\mathcal{Y}$ to $\mathcal{X}$, we specify a mapping $G_{\mathcal{Y}\to\mathcal{X}}$ that seeks to re-express the image $y \in \mathcal{Y}$ by the image $x \in \mathcal{X}$. The latent variable model is of the form:

$$\begin{aligned} &y \sim p_{\text{data}}(y), \\ &x = G_{\mathcal{Y}\to\mathcal{X}}(y; \alpha_{\mathcal{X}}) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 I_D), \end{aligned} \tag{1}$$

where $G_{\mathcal{Y}\to\mathcal{X}}(y; \alpha_{\mathcal{X}})$ is parameterized by an encoder-decoder structure whose parameters are denoted by $\alpha_{\mathcal{X}}$, and ϵ is a Gaussian residual. We assume σ is given and I_D is the D-dimensional identity matrix. In model (1), y is the latent variable of x, because for each $x \in \mathcal{X}$, its version $y \in \mathcal{Y}$ is unobserved. (x and y have the same number of dimensions.)

Given the empirical prior distribution $p_{\text{data}}(y)$ and $q(x|y; \alpha_{\mathcal{X}}) \sim \mathcal{N}(G_{\mathcal{Y}\to\mathcal{X}}(y; \alpha_{\mathcal{X}}), \sigma^2 I_D)$, we can get the joint density $q(x, y; \alpha_{\mathcal{X}}) = p_{\text{data}}(y) q(x|y; \alpha_{\mathcal{X}})$, and the marginal density $q(x; \alpha_{\mathcal{X}}) = \int q(x, y; \alpha_{\mathcal{X}}) dy$. Training the model via maximum likelihood estimation (MLE) requires the prior $p_{\text{data}}(y)$ to be a tractable density (e.g., Gaussian white noise distribution) for calculating the derivative of the data log-likelihood with respect to $\alpha_{\mathcal{X}}$, i.e., $\frac{\partial}{\partial \alpha_{\mathcal{X}}}[\frac{1}{n_x}\sum_{i=1}^{n_x} \log q(x_i; \alpha_{\mathcal{X}})]$, from training examples $\{x_i, i = 1, ..., n_{\mathcal{X}}\}$ in domain $\mathcal{X}$. Due to the unknown prior $p_{\text{data}}(y)$, we can not estimate $\alpha_{\mathcal{X}}$ in model (1) via MLE with an explaining away inference (**?**) or a variational inference (**?**).

3.3 Energy-Based Model as a Teacher

Instead, to avoid the challenging problem of inferring y from x in estimating $\alpha_{\mathcal{X}}$, we can train $G_{\mathcal{Y}\to\mathcal{X}}$ via MCMC teaching by recruiting an energy-based model (EBM), such as the generative ConvNet (**?**), which specifies the distribution of x explicitly up to a normalizing constant:

$$p(x; \theta_{\mathcal{X}}) = \frac{1}{Z(\theta_{\mathcal{X}})} \exp\left[f(x; \theta_{\mathcal{X}})\right] p_0(x), \tag{2}$$

where $Z(\theta_{\mathcal{X}}) = \int \exp\left[f(x; \theta_{\mathcal{X}})\right] p_0(x) dx$ is the intractable normalizing constant, $p_0(x)$ is the Gaussian reference distribution, i.e., $p_0(x) \propto \exp(-||x||^2/2s^2)$. Standard deviation s is a hyperparameter. The energy function $\mathcal{E}(x; \theta_{\mathcal{X}}) = -f(x; \theta_{\mathcal{X}}) + ||x||^2/2s^2$, where f is parameterized by a ConvNet with parameters $\theta_{\mathcal{X}}$. As to learning $p(x; \theta_{\mathcal{X}})$, the maximum likelihood estimator equivalently minimizes the Kullback-Leibler (KL) divergence between the data distribution $p_{\text{data}}(x)$ and the model, $\text{KL}(p_{\text{data}}(x)||p(x; \theta_{\mathcal{X}}))$, over $\theta_{\mathcal{X}}$. The gradient of MLE is given by

$$\begin{aligned} &-\frac{\partial}{\partial \theta_{\mathcal{X}}} \text{KL}(p_{\text{data}}(x)||p(x; \theta_{\mathcal{X}})) \\ =&\text{E}_{p_{\text{data}}}\left[\frac{\partial}{\partial \theta_{\mathcal{X}}} f(x; \theta_{\mathcal{X}})\right] - \text{E}_{p_{\theta_{\mathcal{X}}}}\left[\frac{\partial}{\partial \theta_{\mathcal{X}}} f(x; \theta_{\mathcal{X}})\right], \end{aligned} \tag{3}$$

where $\text{E}_{p_{\theta_{\mathcal{X}}}}$ denotes the expectation with respect to the model $p(x; \theta_{\mathcal{X}})$. In practice, Eq.(3) can be approximated by

$$\Delta(\theta_{\mathcal{X}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial \theta_{\mathcal{X}}} f(x_i; \theta_{\mathcal{X}}) - \frac{1}{n}\sum_{i=1}^{\tilde{n}} \frac{\partial}{\partial \theta_{\mathcal{X}}} f(\tilde{x}_i; \theta_{\mathcal{X}}) \tag{4}$$

where $\{\tilde{x}_i, i = 1, ..., \tilde{n}\}$ are MCMC examples sampled from the current distribution $p(x; \theta_{\mathcal{X}})$. With an EBM model defined on domain $\mathcal{X}$, we can sample x by MCMC, such as Langevin dynamics (**?**), which iterates

$$x_{\tau+1} = x_\tau - \frac{\delta^2}{2}\frac{\partial}{\partial x}\mathcal{E}(x_\tau; \theta_{\mathcal{X}}) + \delta U_\tau, \tag{5}$$

where τ indexes the time step, δ is the step size, and $U_\tau \sim \mathcal{N}(0, I_D)$ is the Gaussian noise term.

In the MCMC teaching process, the EBM $p(x; \theta_{\mathcal{X}})$ can distill its MCMC algorithm to $q(x; \alpha_{\mathcal{X}})$, so that $p(x; \theta_{\mathcal{X}})$ can absorb and accumulate the MCMC transitions in order to reproduce them by one step ancestral sampling. That means, at each MCMC teaching step, $q(x; \alpha_{\mathcal{X}})$ learns to get close to $p(x; \theta_{\mathcal{X}})$, and chases it toward $p_{\text{data}}(x)$. Meanwhile, $q(x; \alpha_{\mathcal{X}})$ will serve as an initializer of the MCMC of $p(x; \theta_{\mathcal{X}})$ for efficient Langevin sampling.

Formally, let $\mathcal{M}_{\theta_{\mathcal{X}}}$ be the Markov transition kernel of l steps of Langevin dynamics that samples from $p(x; \theta_{\mathcal{X}})$, and $\mathcal{M}_{\theta_{\mathcal{X}}} q_{\alpha_{\mathcal{X}}}$ be the marginal distribution obtained by running the Markov transition $\mathcal{M}_{\theta_{\mathcal{X}}}$ initialized by $q_{\alpha_{\mathcal{X}}}$. Training $q_{\alpha_{\mathcal{X}}}$ via MCMC teaching seeks to find $\alpha_{\mathcal{X}}$ at time t to minimize $\text{KL}(\mathcal{M}_{\theta_{\mathcal{X}}} q_{\alpha_{\mathcal{X}}^{(t)}}||q_\alpha)$, which implies a minimization of $\text{KL}(p_{\theta_{\mathcal{X}}}||q_{\alpha_{\mathcal{X}}})$ over $\alpha_{\mathcal{X}}$. Once $\text{KL}(p_{\text{data}}(x)||p(x; \theta_{\mathcal{X}})) \to 0$, then $\text{KL}(p_{\text{data}}(x)||q(x; \alpha_{\mathcal{X}})) \to 0$.

3.4 Cycle-Consistent Cooperative Networks

The energy-based model p and the latent variable model G form a cooperative network. In this paper, to tackle the unsupervised cross-domain translation, we propose a framework consisting of a pair of cooperative networks, i.e.,

$$\begin{aligned} \mathcal{Y} \to \mathcal{X} &: \{G_{\mathcal{Y}\to\mathcal{X}}(y; \alpha_{\mathcal{X}}), p(x; \theta_{\mathcal{X}})\}, \\ \mathcal{X} \to \mathcal{Y} &: \{G_{\mathcal{X}\to\mathcal{Y}}(x; \alpha_{\mathcal{Y}}), p(y; \theta_{\mathcal{Y}})\}, \end{aligned} \tag{6}$$

each of which accounts for one direction of translation between domains $\mathcal{X}$ and $\mathcal{Y}$, and we simultaneously learn them by alternating their MCMC teaching algorithms. To guarantee that each individual input $x \in \mathcal{X}$ or $y \in \mathcal{Y}$ and its translated version $\tilde{y} \in \mathcal{Y}$ or $\tilde{x} \in \mathcal{X}$ are meaningfully paired up, we enforce $G_{\mathcal{Y}\to\mathcal{X}}$ and $G_{\mathcal{X}\to\mathcal{Y}}$ to be inverse functions of each other when training the models, i.e.,

$$\begin{aligned} x &= G_{\mathcal{Y}\to\mathcal{X}}(G_{\mathcal{X}\to\mathcal{Y}}(x;\alpha_{\mathcal{Y}});\alpha_{\mathcal{X}}), \forall x, \\ y &= G_{\mathcal{X}\to\mathcal{Y}}(G_{\mathcal{Y}\to\mathcal{X}}(y;\alpha_{\mathcal{X}});\alpha_{\mathcal{Y}}), \forall y. \end{aligned} \tag{7}$$

We call the proposed framework Cycle-Consistent Cooperative Networks (CycleCoopNets).

3.5 Alternating MCMC Teaching

As illustrated in Figure 1(1), we first sample $y_i \sim p_{\text{data}}(y)$, and then translate via $\hat{x}_i = G_{\mathcal{Y}\to\mathcal{X}}(y_i;\alpha_{\mathcal{X}})$, for $i = 1, ..., \tilde{n}$. Starting from $\{\hat{x}_i, i = 1, ..., \tilde{n}\}$, we run MCMC, e.g., Langevin dynamics, for a finite number of steps toward $p(x;\theta_{\mathcal{X}})$ to obtain $\{\tilde{x}_i, i = 1, ..., \tilde{n}\}$, which are revised versions of $\{\hat{x}_i, i = 1, ..., \tilde{n}\}$. Even though $\{\tilde{x}_i\}$ are cooperatively generated by both $p(x;\theta_{\mathcal{X}})$ and $G_{\mathcal{Y}\to\mathcal{X}}$, they are synthesized examples sampled from $p(x;\theta_{\mathcal{X}})$, and are used to learn $\theta_{\mathcal{X}}$ according to Eq.(4). After updating $\theta_{\mathcal{X}}$, $p(x;\theta_{\mathcal{X}})$ gets close to $p_{\text{data}}(x)$ by fitting all the major modes of $p_{\text{data}}(x)$. See Figure 1(2) for an explanation. The energy-based model $p(x;\theta_{\mathcal{X}})$ then teaches $G_{\mathcal{Y}\to\mathcal{X}}$ by MCMC teaching. The key is that for the cooperatively synthesized examples $\{\tilde{x}_i\}$, their sources $\{y_i\}$ are known. In order to update $\alpha_{\mathcal{X}}$ of $G_{\mathcal{Y}\to\mathcal{X}}$, we treat $\{\tilde{x}_i\}$ as the training data of $G_{\mathcal{Y}\to\mathcal{X}}$. Since these $\{\tilde{x}_i\}$ are obtained by the Langevin dynamics initialized from $\{\hat{x}_i\}$, which are generated by $G_{\mathcal{Y}\to\mathcal{X}}$ with known inputs $\{y_i\}$, we can directly update $\alpha_{\mathcal{X}}$ by learning from the complete data $\{(y_i, \tilde{x}_i), i = 1, ..., \tilde{n}\}$, which is a non-linear regression of $\tilde{x}_i$ on y_i with the objective

$$L_{teach}(\alpha_{\mathcal{X}}) = \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \|\tilde{x}_i - G_{\mathcal{Y}\to\mathcal{X}}(y_i;\alpha_{\mathcal{X}})\|^2. \tag{8}$$

At $\alpha_{\mathcal{X}}^{(t)}$, y_i is mapped to the initial example $\hat{x}_i$. After updating $\alpha_{\mathcal{X}}$, we want y_i to map the revised example $\tilde{x}_i$. That is, we revise $\alpha_{\mathcal{X}}$ to absorb the MCMC transition from $\hat{x}_i$ to $\tilde{x}_i$ for chasing $p(x;\theta_{\mathcal{X}})$. In the meanwhile, we simultaneously train the other mapping from domain $\mathcal{X}$ to $\mathcal{Y}$, i.e., $\{G_{\mathcal{X}\to\mathcal{Y}}, p(y;\theta_{\mathcal{Y}})\}$ in a similar way. To enforce mutual invertibility between $G_{\mathcal{X}\to\mathcal{Y}}$ and $G_{\mathcal{Y}\to\mathcal{X}}$, we can add the following cycle consistency loss while learning $\alpha_{\mathcal{X}}$ and $\alpha_{\mathcal{Y}}$,

$$\begin{aligned} L_{\text{cyc}}(\alpha_{\mathcal{X}}, \alpha_{\mathcal{Y}}) &= \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \|x_i - G_{\mathcal{Y}\to\mathcal{X}}(G_{\mathcal{X}\to\mathcal{Y}}(x_i;\alpha_{\mathcal{Y}});\alpha_{\mathcal{X}})\|_1 \\ &+ \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \|y_i - G_{\mathcal{X}\to\mathcal{Y}}(G_{\mathcal{Y}\to\mathcal{X}}(y_i;\alpha_{\mathcal{X}});\alpha_{\mathcal{Y}})\|_1. \end{aligned} \tag{9}$$

Figure 1(3) is an illustration of the convergence of the algorithm. Algorithm 1 presents a full description of the learning algorithm to train CycleCoopNets.

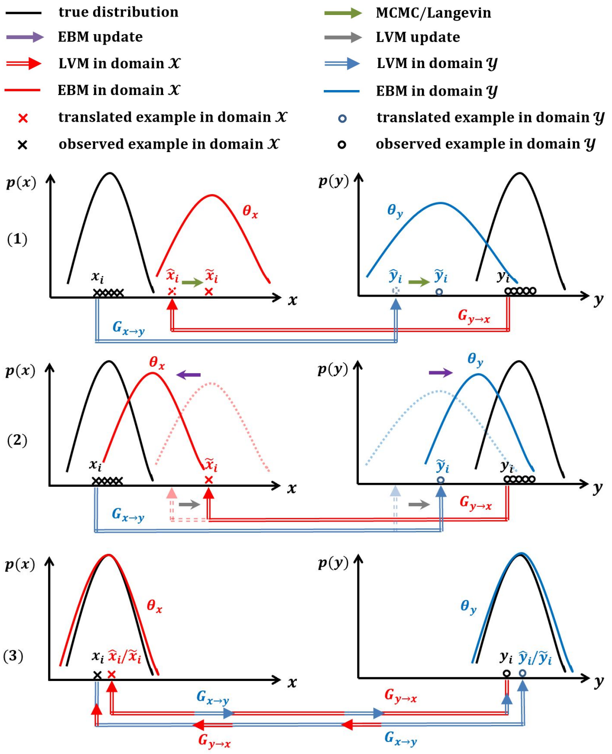

true distribution
MCMC/Langevin
EBM update
LVM update
LVM in domain $\mathcal{X}$
LVM in domain $\mathcal{Y}$
EBM in domain $\mathcal{X}$
EBM in domain $\mathcal{Y}$
translated example in domain $\mathcal{X}$
translated example in domain $\mathcal{Y}$
observed example in domain $\mathcal{X}$
observed example in domain $\mathcal{Y}$
$p(x)$
$p(y)$
(1)
(2)
(3)
θ_x
θ_y
x_i
$\hat{x}_i$
$\tilde{x}_i$
$\hat{y}_i$
$\tilde{y}_i$
y_i
$\hat{x}_i/\tilde{x}_i$
$\hat{y}_i/\tilde{y}_i$
$G_{x\to y}$
$G_{y\to x}$
x
y

Figure 1: An illustration of the alternating MCMC teaching algorithm. (1) cross-domain mapping (2) density shifting (3) mapping shifting with cycle consistency.

4 Generalizing to Unpaired Cross-Domain Image Sequence Translation

We can further generalize the proposed framework to learning a translation between two domains of image sequences where paired examples are unavailable. For example, given an image sequence of Donald Trump's speech, we can translate it to an image sequence of Barack Obama, where the content of Donald Trump is transferred to Barack Obama but the speech is in Donald Trump's style. Such an appearance translation and motion style preservation framework may have a wide range of applications in video manipulation. Even though the translation is made on image sequences, we do not build distributions or mappings on sequence domain. Instead, we rely on translation models defined on image space, and bring in temporal prediction models accounting for temporal information. Therefore, we can make minimal modifications on our current framework discussed in Section 3 for image sequence translation. Suppose we observe two unpaired but ordered image sequences $X = (x_1, x_2, ..., x_t, ...)$ and $Y = (y_1, y_2, ..., y_t, ...)$, and $\forall x_t \in \mathcal{X}, \forall y_t \in \mathcal{Y}$. Each long sequence can be turned into a collection of short sequences with an equal length, i.e., $\{x_{t:t+k}\}_{t=1}^{T_{\mathcal{X}}}$ and $\{y_{t:t+k}\}_{t=1}^{T_{\mathcal{Y}}}$, where $x_{t:t+k} = (x_t, ..., x_{t+k})$ and $y_{t:t+k} = (y_t, ..., y_{t+k})$. The length of each short sequence is $k + 1$. The current framework only learns translation of static image frames between two domains, without considering temporal information existing in each domain.

We need to make the following two modifications to adapt our model to the new task: (i) We learn a temporal prediction model in each domain to predict future image frame given the past image frames in a sequence. Let $R_{\mathcal{X}}$ and $R_{\mathcal{Y}}$ denote temporal prediction models for domains $\mathcal{X}$ and $\mathcal{Y}$ respectively. We learn $R_{\mathcal{X}}$ and $R_{\mathcal{Y}}$ by minimizing

$$\begin{aligned} L_{\text{tp}}(R_{\mathcal{X}}) &= \frac{1}{T_{\mathcal{X}}}\sum_{t=1}^{T_{\mathcal{X}}} \|x_{t+k} - R_{\mathcal{X}}(x_{t:t+k-1})\|_1, \\ L_{\text{tp}}(R_{\mathcal{Y}}) &= \frac{1}{T_{\mathcal{Y}}}\sum_{t=1}^{T_{\mathcal{Y}}} \|y_{t+k} - R_{\mathcal{Y}}(y_{t:t+k-1})\|_1. \end{aligned} \tag{10}$$

(ii) With $R_{\mathcal{X}}$ and $R_{\mathcal{Y}}$, we can modify the loss in Eq. (9) to take into account spatial-temporal information as below

$$\begin{aligned} &L_{\text{st}}(G_{\mathcal{X}\to\mathcal{Y}}, R_{\mathcal{Y}}, G_{\mathcal{Y}\to\mathcal{X}}) \\ =&\frac{1}{T_{\mathcal{X}}}\sum_{t=1}^{T_{\mathcal{X}}} \|x_{t+k} - G_{\mathcal{Y}\to\mathcal{X}}(R_{\mathcal{Y}}(G_{\mathcal{X}\to\mathcal{Y}}(x_{t:t+k-1})))\|_1, \\ &L_{\text{st}}(G_{\mathcal{Y}\to\mathcal{X}}, R_{\mathcal{X}}, G_{\mathcal{X}\to\mathcal{Y}}) \\ =&\frac{1}{T_{\mathcal{Y}}}\sum_{t=1}^{T_{\mathcal{Y}}} \|y_{t+k} - G_{\mathcal{X}\to\mathcal{Y}}(R_{\mathcal{X}}(G_{\mathcal{Y}\to\mathcal{X}}(y_{t:t+k-1})))\|_1, \end{aligned} \tag{11}$$

where, for notation simplicity, we use $G_{\mathcal{X}\to\mathcal{Y}}(x_{t:t+k-1}) = (G_{\mathcal{X}\to\mathcal{Y}}(x_t), ..., G_{\mathcal{X}\to\mathcal{Y}}(x_{t+k-1}))$ and $G_{\mathcal{Y}\to\mathcal{X}}(y_{t:t+k-1}) = (G_{\mathcal{Y}\to\mathcal{X}}(y_t), ..., G_{\mathcal{Y}\to\mathcal{X}}(y_{t+k-1}))$. The final objective of G and R is given by

$$\begin{aligned} \min_{G,R} L(G,R) &= L_{\text{teach}}(G_{\mathcal{Y}\to\mathcal{X}}) + L_{\text{teach}}(G_{\mathcal{X}\to\mathcal{Y}}) \\ &+ \lambda_1 L_{\text{tp}}(R_{\mathcal{X}}) + \lambda_2 L_{\text{st}}(G_{\mathcal{X}\to\mathcal{Y}}, R_{\mathcal{Y}}, G_{\mathcal{Y}\to\mathcal{X}}) \\ &+ \lambda_1 L_{\text{tp}}(R_{\mathcal{Y}}) + \lambda_2 L_{\text{st}}(G_{\mathcal{Y}\to\mathcal{X}}, R_{\mathcal{X}}, G_{\mathcal{X}\to\mathcal{Y}}), \end{aligned} \tag{12}$$

where λ_1 and λ_2 are hyper-parameters. During testing, given a testing image sequence from domain $\mathcal{X}$, $(x_1, x_2, ..., x_t, ...)$, we can translate the whole sequence to domain $\mathcal{Y}$ by mapping each image frame x_t to domain $\mathcal{Y}$ via the learned $G_{\mathcal{X}\to\mathcal{Y}}$, and then revising the result via $p(y; \theta_{\mathcal{Y}})$.

5 Experiments

We perform experiments on the tasks of unsupervised image-to-image translation and image sequence translation to evaluate the CycleCoopNets. The code can be found at the page http://www.stat.ucla.edu/~jxie/CycleCoopNets/.

5.1 Unsupervised Image-to-Image Translation

Implementation We present the network structures of p and G for the mapping from domain $\mathcal{Y}$ to $\mathcal{X}$ and the mapping from domain $\mathcal{X}$ to $\mathcal{Y}$ as below. We use the same bottom-up network structures to parameterize the negative energy functions f for both $p(x; \theta_{\mathcal{X}})$ and $p(y; \theta_{\mathcal{Y}})$, and also the same encoder-decoder structures for $G_{\mathcal{X}\to\mathcal{Y}}$ and $G_{\mathcal{Y}\to\mathcal{X}}$.

Structure of p: Each EBM p has a bottom-up ConvNet structure f that consists of 4 layers of convolutions with numbers of channels $\{64, 128, 256, 512\}$, filter sizes $\{3, 4, 4, 4\}$, and subsampling factors $\{1, 2, 2, 2\}$ at different layers, and one fully connected layer with 100 filters. Leaky ReLU layers are applied between convolutional layers.

Algorithm 1 Alternating MCMC teaching algorithm

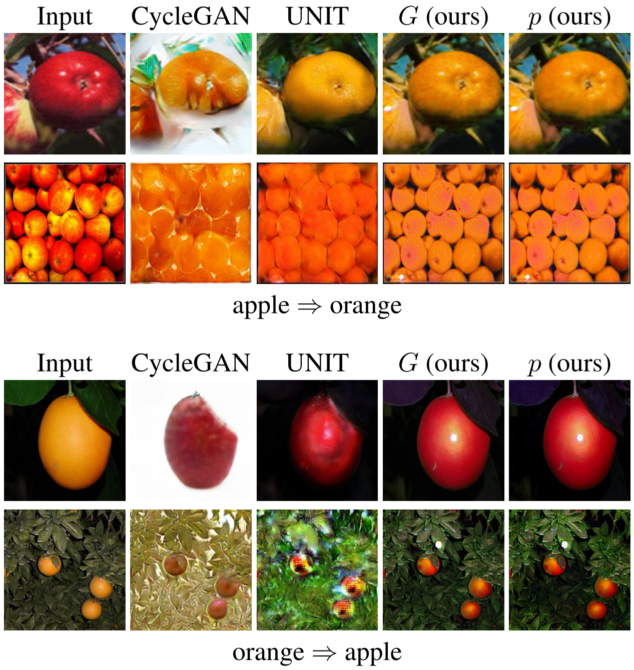
```
Input:
(1) training examples in domain $\mathcal{X}$, $\{x_i, i = 1, ..., n_x\}$, and domain $\mathcal{Y}$, $\{y_i, i = 1, ..., n_y\}$; (2) number of Langevin steps $l$; (3) learning rate $\gamma_{\theta_{\mathcal{X}}}, \gamma_{\theta_{\mathcal{Y}}}, \gamma_{\alpha_{\mathcal{X}}}, \gamma_{\alpha_{\mathcal{Y}}}$; (4) number of learning iterations $T$.
Output:
Estimated parameters $\theta_{\mathcal{X}}, \theta_{\mathcal{Y}}, \alpha_{\mathcal{X}}, \alpha_{\mathcal{Y}}$

Let $t \leftarrow 0$, randomly initialize $\theta_{\mathcal{X}}, \theta_{\mathcal{Y}}, \alpha_{\mathcal{X}}, \alpha_{\mathcal{Y}}$
repeat
    $\{y_i \sim p_{data}(y)\}_{i=1}^{\tilde{n}}$
    $\{\hat{x}_i = G_{\mathcal{Y}\to\mathcal{X}}(y_i; \alpha_{\mathcal{X}})\}_{i=1}^{\tilde{n}}$
    $\{x_i \sim p_{data}(x)\}_{i=1}^{\tilde{n}}$
    $\{\hat{y}_i = G_{\mathcal{X}\to\mathcal{Y}}(x_i; \alpha_{\mathcal{Y}})\}_{i=1}^{\tilde{n}}$.
    Starting from $\{\hat{x}_i\}_{i=1}^{\tilde{n}}$, run $l$ steps of Langevin revision in Eq. (5) to obtain $\{\tilde{x}_i\}_{i=1}^{\tilde{n}}$.
    Starting from $\{\hat{y}_i\}_{i=1}^{\tilde{n}}$, run $l$ steps of Langevin revision in Eq. (5) to obtain $\{\tilde{y}_i\}_{i=1}^{\tilde{n}}$.
    Given $\{x_i\}_{i=1}^{\tilde{n}}$ and $\{\tilde{x}_i\}_{i=1}^{\tilde{n}}$, update $\theta_{\mathcal{X}}^{(t+1)} = \theta_{\mathcal{X}}^{(t)} + \gamma_{\theta_{\mathcal{X}}}\Delta(\theta_{\mathcal{X}}^{(t)})$, where $\Delta(\theta_{\mathcal{X}}^{(t)})$ is computed by Eq. (4).
    Given $\{y_i\}_{i=1}^{\tilde{n}}$ and $\{\tilde{y}_i\}_{i=1}^{\tilde{n}}$, update $\theta_{\mathcal{Y}}^{(t+1)} = \theta_{\mathcal{Y}}^{(t)} + \gamma_{\theta_{\mathcal{Y}}}\Delta(\theta_{\mathcal{Y}}^{(t)})$, where $\Delta(\theta_{\mathcal{Y}}^{(t)})$ is computed by Eq. (4).
    Given $\{y_i\}_{i=1}^{\tilde{n}}$, $\{x_i\}_{i=1}^{\tilde{n}}$ and $\{\tilde{x}_i\}_{i=1}^{\tilde{n}}$, update $\alpha_{\mathcal{X}}^{(t+1)} = \alpha_{\mathcal{X}}^{(t)} - \gamma_{\alpha_{\mathcal{X}}}\nabla(\alpha_{\mathcal{X}}^{(t)})$, where $\nabla(\alpha_{\mathcal{X}}^{(t)})$ is the gradient of loss functions in Eq. (8) and Eq. (9).
    Given $\{x_i\}_{i=1}^{\tilde{n}}$, $\{y_i\}_{i=1}^{\tilde{n}}$ and $\{\tilde{y}_i\}_{i=1}^{\tilde{n}}$, update $\alpha_{\mathcal{Y}}^{(t+1)} = \alpha_{\mathcal{Y}}^{(t)} - \gamma_{\alpha_{\mathcal{Y}}}\nabla(\alpha_{\mathcal{Y}}^{(t)})$, where $\nabla(\alpha_{\mathcal{Y}}^{(t)})$ is the gradient of loss functions in Eq. (8) and Eq. (9).
    Let $t \leftarrow t + 1$
until $t = T$
```

Structure of G. We adopt the architecture from **?** for G. The CycleGAN also uses the same architecture. We use 9 residual blocks. We only replace the instance normalization by the batch normalization in this architecture for art style transfer.

Hyperparameters We use 20 Langevin steps for season transfer and 15 steps for other experiments. The Langevin step size is 0.002. The standard deviation s of the reference distribution of the EBM is 0.016. We use Adam (**?**) for optimization with a learning rate 0.0002. The hyperparameter that weighs the relative contribution of the cycle consistency loss term in the total loss is $\lambda_{cyc} = 9$. The batch size is 1. The number of parallel chains is 1.

Object transfiguration We train our model to translate one object category from ImageNet (**?**) to another. Each category has roughly 1,000 training examples. Figure 2 displays some testing results of an example of object transfiguration between categories apple and orange. Each panel shows one direction of translation, in which the first column displays the input images, the second and the third columns show the results obtained by two baseline methods Cycle-



apple ⇒ orange

orange ⇒ apple

Figure 2: Object transfiguration. The top panel displays the translation from apples to oranges, and the bottom panel displays the translation from oranges to apples. For each panel, the first column shows the input images, and the rest show the translated results obtained by different models.

GAN (?) and UNIT (?), respectively. The last two columns show the results achieved by our model, where the fourth column displays the results generated by G without using p's MCMC revision, and the fifth column displays those obtained by p's MCMC, which is initialized by G. These qualitative results suggest that the proposed framework can be successfully applied to unsupervised image-to-image translation. Moreover, for each pair of G and p, G traces p, and p traces p_{data}, thus G will eventually get close to p. The results shown in Figure 2 verify this fact in the sense that the final results generated by G (fourth column) and p (fifth column) look almost the same and indistinguishable.

To quantitatively evaluate the performance of the proposed model, we use the Fréchet Inception Distance (FID) (?) to measure the similarity between the translated distribution and the target distribution. This distribution matching metric can indicate to what extent the input images in one domain are translated to the other domain. We compute the FID score between the set of translated images and the set of observed images in the target domain. We use the activations from the last average pooling layer of the Inception-V3 (?) model, which is pretrained on ImageNet (?) for classification, as features of each image for computing the FID. A lower FID score is desired because it corresponds to a higher similarity between the target distribution and the translated one.

methods	**apple ⇒ orange**		**orange ⇒ apple**	
	FID ↓	DIPD ↓	FID↓	DIPD ↓
CycleGAN	160.78	1.75	143.87	1.73
UNIT	170.66	1.58	122.04	1.62
$G(l = 15)$	158.66	1.28	119.27	1.34
$p(l = 15)$	**154.58**	**1.23**	**118.82**	**1.25**
$p(l = 1)$	192.60	1.43	143.00	1.42
$p(l = 5)$	166.41	1.43	170.38	1.40
$p(l = 10)$	189.60	1.32	141.60	1.32

Table 1: Quantitative evaluation on apple⇔orange dataset with respect to Fréchet Inception Distance (FID) and Domain-invariant Perceptual Distance (DIPD). The top two rows show the results of CycleGAN and UNIT. The middle two rows show the results of G and p, where l is the number of Langevin steps. The last three rows show performances of the models with different numbers of Langevin steps.

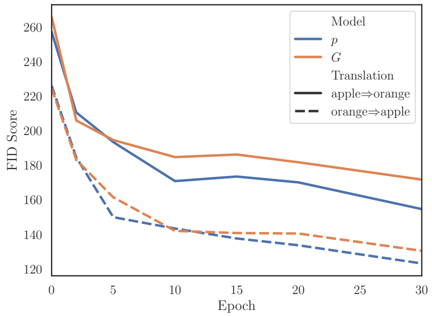
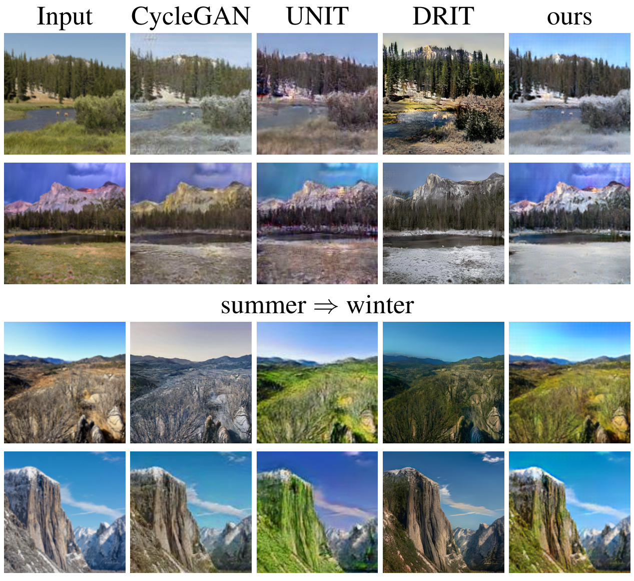


Figure 3: Learning curves that represent FID scores of the translated images (apple ⇒ orange or orange ⇒ apple) over epochs. The blue and yellow curves represent the result achieved by p and G respectively. The results suggest that p refines the results provided by G.

We additionally evaluate our results by the domain-invariant perceptual distance (DIPD) (?), which can be used to measure the content preservation in unsupervised image-to-image translation. According to **?**, the DIPD is given by the L2-distance between the normalized VGG (?) Conv5 features of the input image and the translated image. We expect the content in the input image is preserved in the translated image, thus a lower DIPD is desired.

As shown in Table 1, the proposed framework outperforms the baseline models CycleGAN and UNIT on both metrics FID and DIPD. We also study the effect of the number of Langevin steps used in the model. Performances of our models with different numbers of Langevin steps are evaluated by FID and DIPD in the last 3 rows of Table 1.

Figure 3 shows the learning curves that represent FID scores of the translated images obtained by G (orange curves) and p (blue curves) over training epochs. The solid curves represent the results obtained on the translation from apples to oranges, while the dashed curves represent the results for the other direction of translation. We observe improvements in quality of the results in terms of FID score, as the learning algorithm proceeds. We also observe the MCMC refinement effect of p on G in the learning curves.



summer ⇒ winter

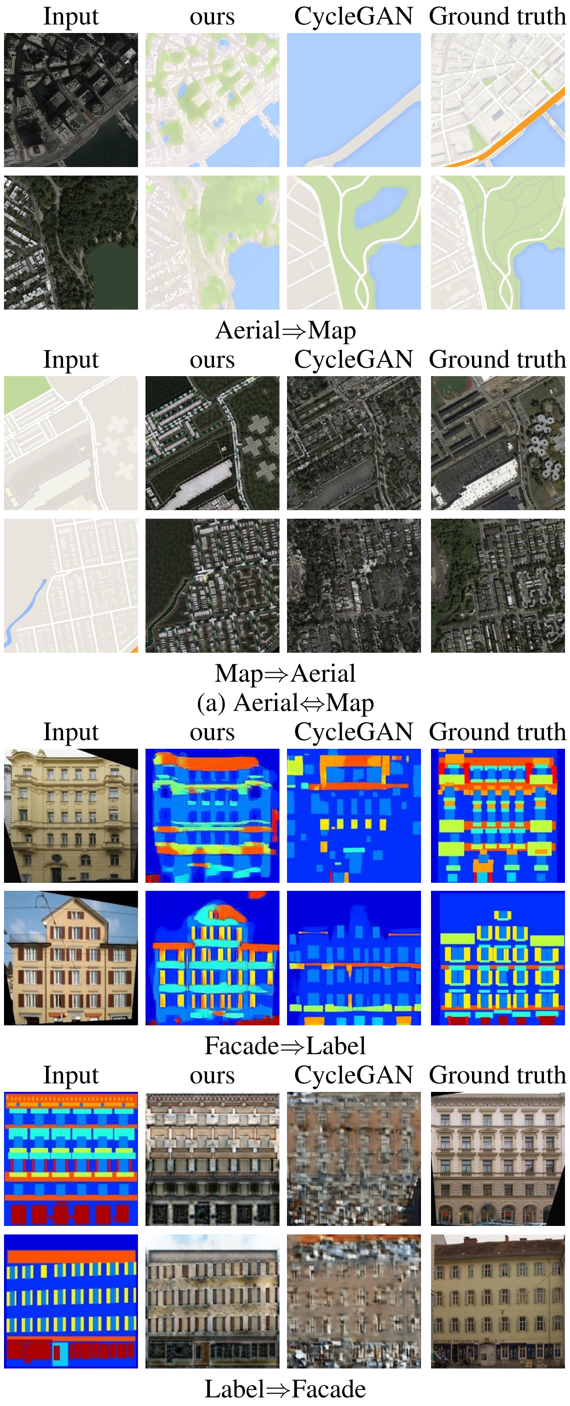
winter ⇒ summer

Figure 4: Example results of season transfer on summer and winter Yosemite photos from Flickr. For each penal, the first column shows some examples of testing input images, and the rest columns display the images "translated" by CycleGAN, UNIT, DRIT and our method, respectively.

Season transfer We train our model on 854 winter photos and 1,273 summer photos of Yosemite that are used in **?** for season transfer. Figure 4 shows some qualitative results and compares against three baseline methods, including CycleGAN, UNIT and DRIT (**?**). Our model obtains more realistic translation results compared with other baseline methods.

Dataset	Model	↑ L⇒R	↑ R⇒L
Aerial⇔Map	CycleGAN	21.59	12.67
	AlignFlow(mle)	19.47	13.60
	AlignFlow(adv)	20.16	**15.17**
	Ours	**22.29**	14.50
Facade⇔Label	CycleGAN	6.68	7.61
	AlignFlow(mle)	6.47	8.26
	AlignFlow(adv)	7.74	11.74
	Ours	**9.34**	**11.93**

Table 2: Quantitative evaluation in terms of PSNR on datasets Aerial⇔Map and Facade⇔Label. (adv: adversarial learning; mle: maximum likelihood estimation)



Aerial⇒Map

Map⇒Aerial

(a) Aerial⇔Map

Facade⇒Label

Label⇒Facade

(b) Facade⇔Label

Figure 5: Qualitative results of unpaired image-to-image translations on datasets (a) aerial⇔ map (b) facade ⇔ label.

Translation between photo image and semantic label image We evaluate the proposed framework on two image-to-image translation datasets: (a) Aerial⇔Map (**?**) and (b) Facade⇔label (**?**). These two datasets provide one-to-one paired images that are originally used for supervised image-to-image translation in **?**. In this experiment, we train our model on the datasets in an unsupervised manner, where the correspondence information between two image domains is omitted. We only use this correspondence information at the testing stage to compute the similarity between the generated images and the corresponding ground truths for quantitative evaluation. Table 2 shows a comparison of our method and some baselines, which include CycleGAN and Align-

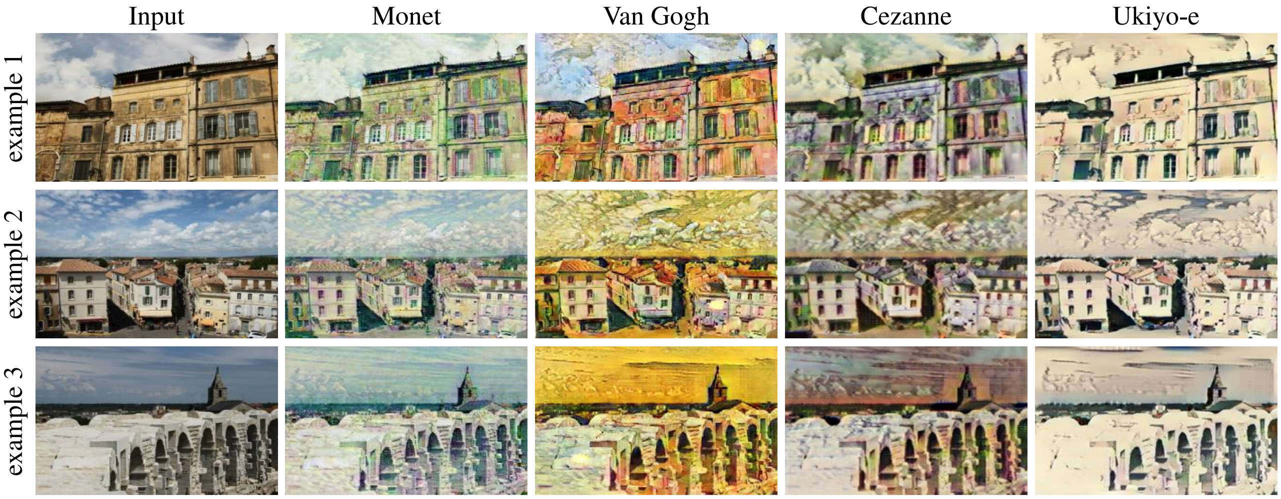


Figure 6: Collection style transfer from photo realistic images to artistic styles.

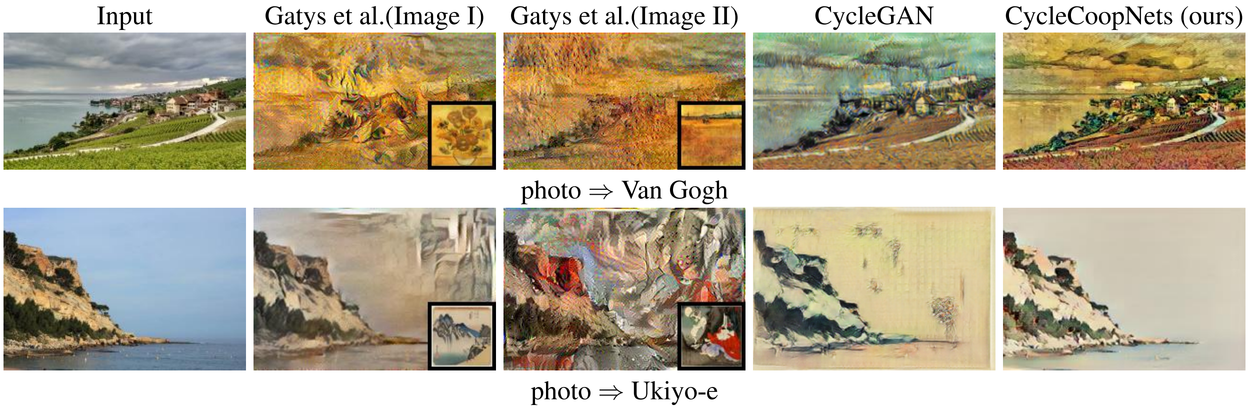


photo ⇒ Van Gogh

photo ⇒ Ukiyo-e

Figure 7: We compare our framework with style transfer method using neural network (**?**) on photo stylization. Each row represents one example, where the first column shows the input image, the second and the third columns show results from **?** using two different representative artworks as style images, the fourth column displays the result of CycleGAN, and the last one is the result by our method CycleCoopNets.

Flow (**?**). AlignFlow is a generative framework that uses normalizing flows for unsupervised image-to-image translation. We consider two types of training for the AlignFlow. One is based on maximum likelihood estimation, while the other is based on adversarial learning. We measure the similarity between two images via peak signal-to-noise ratio (PSNR), which is a suitable metric for evaluating datasets with one-to-one paring information. Figure 5 displays some qualitative results for both datasets. Our method shows comparable results with the baselines.

Art style transfer We evaluate our model on collection style transfer. We learn to translate landscape photographs into art paintings in the styles of Monet, Van Gogh, Cezanne and Ukiyo-e. The collections of landscape photographs are downloaded from Flickr and WikiArt and used in **?**. We train a model between the photograph collection and each of the art collections to obtain the translator from photograph domain to painting domain. Figure 6 displays some results. Each column represents one artistic style.

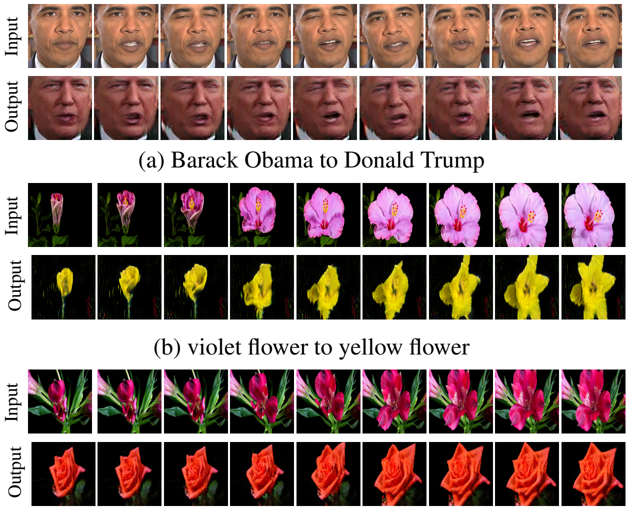
In Figure 7, we compare our model with neural style transfer (**?**) and CycleGAN on photo stylization. Different from our method, **?** requires an image that specifies the target style to stylize an input photo image. Different rows show experiments with different target artistic styles. For each row, the input photo image is displayed in the first column, and we choose two representative artworks from the artistic collection as the style images for **?** and show their results in the second and third columns, respectively. CycleGAN and our method can stylize photos based on the style of the entire artistic collection, whose results are respectively shown in the last two columns. We find that nerual style transfer method (**?**) is difficult to generate meaningful results, while our method succeeds to produce meaningful ones that have a similar style to the target domain, which are comparable with those obtained by CycleGAN.

Time complexity We highlight three points regarding the time complexity. (1) Although learning EBMs involves MCMC, each encoder-decoder G in our framework serves to initialize the MCMC process, so that we only need a few steps of MCMC at each iteration. (2) Our MCMC method is the Langevin dynamics, which is a gradient-based algorithm, which means we only need to compute the gradient of the ConvNet-parameterized energy function with respect to the image. This can be efficiently accomplished by back-propagation due to the differentiability of the ConvNet. Other sampling methods or parametrization methods might not have such a convenience. (3) For implementation, we use TensorFlow as our framework and build the l-step MCMC process as a static computational graph that enables an efficient offline sampling. In all, the whole proposed framework is efficient and can be scaled up for large datasets with current PCs and GPUs. Taking the task of style transfer on the VanGogh2photo dataset (roughly 6,200 training examples) as an example, for training images of size 256 × 256, our training time is roughly 0.80 seconds per iteration with 30 Langevin steps, while the CycleGAN takes 0.28 seconds per iteration. The execution time is recorded in a PC with an Intel i7-6700k CPU and a Titan Xp GPU.

5.2 Unsupervised Image Sequence Translation

We test our framework for image sequence translation. We use a U-Net structure as the temporal prediction model R, which takes as input a concatenation of two consecutive image frames in the past and predicts the next future frame. The U-Net structure follows the same design as the one used in **?**. The EBM p has a bottom-up ConvNet structure f that consists of 3 layers of convolutions with numbers of channels $\{64, 128, 256\}$, filter sizes $\{5, 3, 3\}$, and subsampling factors $\{2, 2, 1\}$ at different layers, and one fully connected layer with 10 filters. Leaky ReLU layers are used between convolutional layers. The encoder-decoder G is the same as the one we use in Section 5.1. The step size for Langevin is 0.02. The number of Langevin steps is 15 for each EBM. We adopt Adam for optimization with a learning rate 0.0002. We set $\lambda_1 = 9$ and $\lambda_2 = 9$ in Eq.(12). The mini-batch size is 1, and the number of parallel chains is 1.

Figure 8 (a) shows an example of face-to-face translation from Barack Obama to Donald Trump. The first row shows some examples of image frames in the input image sequence, while the second row shows the corresponding image frames of the translated image sequence. For this experiment, the training sequences of faces are from **?**, in which the faces are extracted from publicly available videos of public figures, based on keypoints detected using the OpenPose library (**?**). The size of the image frame is 128 × 128 pixels. Figure 8 (b) and (c) show two examples of flower-to-flower translation. All training sequences are about blooming of different flowers. The image frames in each training sequence are ordered but all the sequences are neither synchronous nor aligned. The setting of the experiment is the same as the one of face-to-face translation. The results show that our framework can learn reasonable translation between two sequence domains without synchronous or aligned image frames. In particular, the translated sequences preserve the motion styles of the input sequences and only change their contents or appearances.



(a) Barack Obama to Donald Trump

(b) violet flower to yellow flower

(c) purple flower to red flower

Figure 8: Image sequence translation. We translate (a) Barack Obama's facial motion to Donald Trump. (b) the blooming of a violet flower to a yellow flower. (c) the blooming of a purple flower to a red flower. For each case, the first row displays some image frames of the input sequence, and the second row shows the corresponding translated results.

6 Conclusion

This paper studies unsupervised cross-domain translation problem based on a cooperative learning scheme. Our framework includes two cooperative networks, each of which consists of an energy-based model and a latent variable model to account for one domain distribution. We propose the alternating MCMC teaching algorithm to simultaneously train the two cooperative networks for maximum likelihood and cycle consistency. Experiments show that the proposed framework can be useful for different unsupervised cross-domain translation tasks.

Acknowledgements

The work is supported by NSF DMS-2015577, DARPA XAI N66001-17-2-4029, ARO W911NF1810296, ONR MURI N00014-16-1-2007, and XSEDE grant ASC180018. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML)*, 214–223.

Arora, S.; Risteski, A.; and Zhang, Y. 2018. Do GANs learn the distribution? Some theory and empirics. In *International Conference on Learning Representations (ICLR)*.

Bansal, A.; Ma, S.; Ramanan, D.; and Sheikh, Y. 2018. Recycle-GAN: Unsupervised video retargeting. In *European Conference on Computer Vision (ECCV)*, 119–135.

Brock, A.; Donahue, J.; and Simonyan, K. 2018. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*.

Cao, Z.; Simon, T.; Wei, S.-E.; and Sheikh, Y. 2017. Realtime multi-person 2D pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7291–7299.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255.

Denton, E. L.; Chintala, S.; Fergus, R.; et al. 2015. Deep generative image models using a Laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, 1486–1494.

Du, Y.; and Mordatch, I. 2019. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 3608–3618.

Gao, R.; Lu, Y.; Zhou, J.; Zhu, S.-C.; and Wu, Y. N. 2018. Learning generative ConvNets via multi-grid modeling and sampling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9155–9164.

Gatys, L.; Ecker, A.; and Bethge, M. 2016a. A neural algorithm of artistic style. *Journal of Vision* 16(12): 326–326.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016b. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2414–2423.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2672–2680.

Grover, A.; Chute, C.; Shu, R.; Cao, Z.; and Ermon, S. 2020. AlignFlow: Cycle consistent learning from multiple domains via normalizing flows. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, 4028–4035.

Han, T.; Lu, Y.; Zhu, S.-C.; and Wu, Y. N. 2017. Alternating back-propagation for generator network. In *The Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, 1976–1984.

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 (NIPS)*, 6626–6637.

Huang, X.; Liu, M.-Y.; Belongie, S.; and Kautz, J. 2018. Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision (ECCV)*, 172–189.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1125–1134.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 694–711.

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

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

Lee, H.-Y.; Tseng, H.-Y.; Mao, Q.; Huang, J.-B.; Lu, Y.-D.; Singh, M.; and Yang, M.-H. 2020. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision (IJCV)* 128(10): 2402–2417.

Liu, M.-Y.; Breuel, T.; and Kautz, J. 2017. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems (NIPS)*, 700–708.

Luan, F.; Paris, S.; Shechtman, E.; and Bala, K. 2017. Deep photo style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4990–4998.

Neal, R. M. 2011. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo* 2.

Nijkamp, E.; Hill, M.; Han, T.; Zhu, S.-C.; and Wu, Y. N. 2020. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, 5272–5280.

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 (NeurIPS)*, 5232–5242.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations (ICLR)* .

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826.

Tyleček, R.; and Šára, R. 2013. Spatial pattern templates for recognition of objects with regular structure. In *German Conference on Pattern Recognition (GCPR)*, 364–374.

Ulyanov, D.; Lebedev, V.; Vedaldi, A.; and Lempitsky, V. 2016. Texture networks: Feed-forward synthesis of textures and stylized images. In *International Conference on Machine Learning (ICML)*, 1349–1357.

Xie, J.; Lu, Y.; Gao, R.; and Wu, Y. N. 2018a. Cooperative learning of energy-based model and latent variable model via MCMC teaching. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 4292–4301.

Xie, J.; Lu, Y.; Gao, R.; Zhu, S.-C.; and Wu, Y. N. 2018b. Cooperative training of descriptor and generator networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* .

Xie, J.; Lu, Y.; Zhu, S.-C.; and Wu, Y. N. 2016. A theory of generative ConvNet. In *International Conference on Machine Learning (ICML)*.

Xie, J.; Xu, Y.; Zheng, Z.; Zhu, S.-C.; and Wu, Y. N. 2020a. Generative PointNet: Energy-based learning on unordered point sets for 3D generation, reconstruction and classification. *arXiv preprint arXiv:2004.01301* .

Xie, J.; Zheng, Z.; Fang, X.; Zhu, S.-C.; and Wu, Y. N. 2019. Cooperative training of fast thinking initializer and slow thinking solver for conditional learning. *arXiv preprint arXiv:1902.02812* .

Xie, J.; Zheng, Z.; Gao, R.; Wang, W.; Zhu, S.-C.; and Wu, Y. N. 2018c. Learning descriptor networks for 3D shape synthesis and analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8629–8638.

Xie, J.; Zheng, Z.; Gao, R.; Wang, W.; Zhu, S.-C.; and Wu, Y. N. 2020b. Generative VoxelNet: Learning energy-based models for 3D shape synthesis and analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)* .

Xie, J.; Zheng, Z.; and Li, P. 2020. Learning energy-based model with variational auto-encoder as amortized sampler. *arXiv preprint arXiv:2012.14936* .

Xie, J.; Zhu, S.-C.; and Wu, Y. N. 2017. Synthesizing dynamic patterns by spatial-temporal generative ConvNet. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7093–7101.

Xie, J.; Zhu, S.-C.; and Wu, Y. N. 2019. Learning energy-based spatial-temporal generative ConvNets for dynamic patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* .

Zhang, C.; Zhu, Y.; and Zhu, S.-C. 2018. MetaStyle: Three-way trade-off among speed, flexibility, and quality in neural style transfer. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, 1254–1261.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE Conference on Computer Vision (CVPR)*, 2223–2232.