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], 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
jianwen@ucla.edu, z.zheng@ucla.edu, xiaolinf@csail.mit.edu, sczhu@stat.ucla.edu, ywu@stat.ucla.edu

Abstract

This paper studies unsupervised cross-domain translation problem by proposing a generative framework that represents each domain via a cooperative network that consists of a pair of energy-based model and latent variable model. The use of cooperative network enables maximum likelihood estimation of the domain distribution 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, the latent variable model parameterized by an encoder-decoder maps an individual example from source domain to the target domain, while the energy-based model further refines the initial result by MCMC such that the final translated results match to the examples in the target domain in terms of some statistical properties represented by the learned energy function. The proposed framework simultaneously learns and aligns two cooperative networks, accounting for two opposite directions of mappings between two domains, by alternating MCMC teaching for the purpose of building up cross-domain correspondence. Experiments show that the proposed framework can be 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 last few years on numerous computer vision and computer graphics tasks which require translating a datapoint 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 case it is even impossible. For example, learning to translate a photo to a Van Gogh painting style requires a plenty of training pairs of real scene photos and corresponding paintings. 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 Network (GAN) (Goodfellow et al. 2014; Arjovsky, Chintala, and Bottou 2017) in image generation (Radford, Metz, and Chintala 2016; Denton et al. 2015; Brock, Donahue, and Simonyan 2018), researchers have proposed unsupervised cross-domain translation networks based on GANs and obtained compelling results (Zhu et al. 2017; Liu, Breuel, and Kautz 2017; Huang et al. 2018). From a probabilistic modeling perspective, to learn a data probability distribution, instead of maximizing the likelihood, GANs introduce the concept of adversarial learning between a generator and a discriminator. The generator is the desired implicit data distribution that maps a Gaussian prior on a latent space to target data space via non-linear transformations, while the discriminator aims at distinguishing the real data and the "fake" data 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. (Xie et al. 2016, 2018c; Xie, Zhu, and Wu 2017, 2019; Gao et al. 2018; Nijkamp et al. 2019) suggest that highly realistic examples can be generated by Markov chain Monte Carlo (MCMC) sampling from the learned EBMs. Cooperative networks (CoopNets) (Xie et al. 2018b) proposes to learn the EBM simultaneously with a generator model in a cooperative learning scheme, where the generator plays a role of a fast sampler to initialize the MCMC sampling of the EBM, while the EBM teaches the generator via a finite step MCMC. In 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.

Compared to GANs, MCMC-based cooperative learning has a number of conceptual advantages for modeling data distribution: (1) **Avoid mode collapse**. Due to the fact that both EBM and generator in the cooperative learning frame-

$*$ Equal contributions

work are trained generatively by maximum likelihood, it is stable and does not encounter mode collapse issue (Xie et al. 2018b) (2) **MCMC refinement**. Although both GAN and CoopNets involves two sub-models, their interactions are essentially different. GAN will discard the discriminator once the generator is trained, while for CoopNets, in both training and testing stage, the EBM enables a refinement for the generator by MCMC sampling. (3) **Fast-thinking and slow-thinking**. Solving a challenging problem usually requires an iterative algorithm. This amounts to slow thinking. The iterative algorithm usually requires a good initialization to jumpstart it so that it can converge quickly. The initialization amounts to fast thinking. For instance, reasoning and planning usually require iterative search or optimization, which can be initialized by a learned computation in the form of a neural network. Thus integrating fast thinking initialization and slow thinking sampling or optimization is very compelling. The cooperative learning corresponds to a fast-thinking and slow-thinking system (Xie et al. 2019), where the generator serves as a fast-thinking initializer and the EBM serves as a slow-thinking solver. The problem they cooperatively solve in this task is translation.

Our framework for unsupervised cross-domain translation is based on cooperative learning scheme. We first propose to represent each domain by a cooperative network that includes an energy-based model and a latent variable model, where both of these two models are trained via MCMC teaching. Specifically, the latent variable model maps examples from one domain to another, while the EBM further refines the initial translated results by MCMC so that the final translated results match to the examples in the target domain in terms of some statistical properties defined by the EBM. By simultaneously learning and aligning two cooperative network, each of which accounts for one direction of mapping between two domains, by alternating MCMC teaching, we can achieve a novel framework for unsupervised cross-domain translation.

In light of the impressive results in the task of unsupervised image-to-image translation, including object transfiguration, season transfer, and art style transfer, as well as image sequence translation, our work offers a statistical understanding of how MCMC-based cooperative learning can be applied to learning instance-level conditional mapping between two domains from unpaired data.

The contributions of our paper are four-folds:

1. We present a novel framework to study unsupervised cross-domain translation, where two groups of energy-based model and latent variable model that represent two different domain distributions respectively, are simultaneously learned and aligned by alternating MCMC teaching.
2. We provide a theoretical understanding and convergence analysis of the proposed model and learning algorithm. (see supplementary)
3. We apply our framework to a wide range of applications of unsupervised cross-domain translation, including object transfiguration, season transfer, and art style transfer. We show that our model achieves comparing results compared with GAN-based models.
4. We further generalize our framework to the task of unsupervised video retargeting.

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 (Isola et al. 2017) CycleGAN (Zhu et al. 2017) and RecycleGAN (Bansal et al. 2018). By generalizing the original unconditioned GANs to the conditioned scenarios, Pix2Pix is a framework for supervised conditional learning, which has achieved impressive results on paired image-to-image translation tasks, such as colorization, sketch to photo synthesis, etc. CycleGAN is proposed to learn bidirectional translation between two image domains in the absence of paired examples, by jointly training two GANs, each of them accounting for one directional translation, and enforcing cycle-consistency between them. RecycleGAN (Bansal et al. 2018) is designed to translate between two domains of image sequences without paired examples by enforcing spatiotemporal consistency. Note that our work does not belong to adversarial learning but energy-based learning for unpaired data translation.

Energy-based synthesis. (Xie et al. 2016) proposes to adopt a modern convolutional neural network to parameterize the energy function of the energy-based model and learns 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 (Xie et al. 2016; Du and Mordatch 2019; Nijkamp et al. 2020), videos (Xie, Zhu, and Wu 2017, 2019), 3D voxels (Xie et al. 2018c, 2020b) and point clouds (Xie et al. 2020a). (Gao et al. 2018) learns the model with multigid sampling and (Nijkamp et al. 2019) learns the model with short-run MCMC. Our paper is related to energy-based synthesis, because we train both energy-based models and latent variable models for the task of unsupervised cross-domain translation.

Cooperative learning. (Xie et al. 2018a,b) proposes the generative cooperative network (CoopNets) that trains an energy-based model, such as the generative ConvNet model, with the help of a generator network serving as an approximate direct sampler. The energy-based generative ConvNet distills its knowledge to the generator via MCMC, and this is called MCMC teaching. (Xie et al. 2019) further proposes the conditional CoopNets model for supervised image-to-image translation. Unlike the above approaches, our paper proposes to simultaneously learn and align two cooperative networks, where each of them accounts for one direction of mapping between domains, by alternating MCMC teaching for unpaired data translation.

Style transfer using neural networks. (Gatys, Ecker, and Bethge 2016b) first proposes 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 (Simonyan and Zisserman 2015) features. Other works include (Johnson, Alahi, and Fei-Fei 2016; Ulyanov et al. 2016; Luan et al. 2017; Zhang, Zhu, and Zhu 2018). 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 data domains, say $\mathcal{X}$ and $\mathcal{Y}$, and data collections from these two different 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. Let $p_{\text{data}}(x)$ and $p_{\text{data}}(y)$ denote the unknown data distributions for these two domains. Without instance-level correspondence between two collections, we want to learn transition 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 domain $\mathcal{Y}$ by the image x in domain $\mathcal{X}$. The latent variable model is of the following form:

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

where $G_{\mathcal{Y}\to\mathcal{X}}(y; \alpha_{\mathcal{X}})$ is paramertized by an encoder-decoder network 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 domain $\mathcal{X}$, its version y in domain $\mathcal{Y}$ is unobserved. x has the same number of dimensions as y.

Given the prior distribution $p_{\text{data}}(y)$ and $q(x|y) \sim \text{N}(G_{\mathcal{Y}\to\mathcal{X}}(y; \alpha_{\mathcal{X}}), \sigma^2 I_D)$, the joint density $q(x, y; \alpha_{\mathcal{X}}) = p_{\text{data}}(y) q(x|y; \alpha_{\mathcal{X}})$, and the marginal density is $q(x; \alpha_{\mathcal{X}}) = \int q(x, y; \alpha_{\mathcal{X}}) dy$. The model trained with maximum likelihood estimation (MLE) requires a prior $p_{\text{data}}(y)$ with tractable density (e.g., Gaussian white noise distribution) to calculate the derivative of the data log-likelihood with respect to $\alpha_{\mathcal{X}}$, i.e., $\frac{\partial}{\partial \alpha_{\mathcal{X}}}[\frac{1}{n}\sum_{i=1}^{n} \log q(x_i; \alpha_{\mathcal{X}})]$, 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 (Han et al. 2017) or a variational inference (Kingma and Welling 2014).

3.3 Energy-based model as a critic.

Instead, to avoid the challenging problem of inferring y from x, we can train $G_{\mathcal{Y}\to\mathcal{X}}$ via MCMC teaching by recruiting an energy-based model (EBM) (Xie et al. 2016), (Xie, Zhu, and Wu 2017) that specifies the distribution of x explicitly up to a normalizing constant:

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

where $-f(x; \theta_{\mathcal{X}})$ defines the energy of x and is parametrized by a bottom-up deep neural network with parameters $\theta_{\mathcal{X}}$, and $Z(\theta_{\mathcal{X}}) = \int \exp\left[f(x; \theta_{\mathcal{X}})\right] dx$ is the intractable normalizing constant. To learn $p(x; \theta_{\mathcal{X}})$, the maximum likelihood estimator equivalently minimizes the Kullback-Leibler divergences 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

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

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

$$\frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial \theta} f(x_i; \theta) - \frac{1}{n}\sum_{i=1}^{\tilde{n}} \frac{\partial}{\partial \theta} f(\tilde{x}_i; \theta), \tag{4}$$

where $\tilde{x}_i$ 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

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

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

Let $\mathcal{M}_\theta$ be the Markov transition kernel of l steps of Langevin dynamics that samples from $p(x; \theta)$, and $\mathcal{M}_\theta q$ be the marginal distribution obtained by running the Markov transition $\mathcal{M}_\theta$ from distribution q. The EBM $p(x; \theta_{\mathcal{X}})$ can distill its MCMC algorithm to $q(x; \alpha_{\mathcal{X}})$ through MCMC teaching, which seeks to find α at time t to minimize $\text{KL}(\mathcal{M}_\theta q_{\alpha^{(t)}} | q_\alpha)$. That is, $q(x; \theta_{\mathcal{X}})$ gets close to $p(x; \theta_{\mathcal{X}})$.

3.4 Alternating MCMC teaching

In this paper, we propose a framework to simultaneously learn and align two pairs of EBM and latent variable model, i.e., $\{p(x; \theta_{\mathcal{X}}), G_{\mathcal{Y}\to\mathcal{X}}(y; \alpha_{\mathcal{X}})\}$ and $\{p(y; \theta_{\mathcal{Y}}), G_{\mathcal{X}\to\mathcal{Y}}(x; \alpha_{\mathcal{Y}})\}$ for cross-domain translation. Each pair of models is trained via MCMC teaching to form a one-way mapping, i.e., $p(y; \theta_{\mathcal{Y}})$ and $G_{\mathcal{X}\to\mathcal{Y}}$ forms a translation from domain $\mathcal{X}$ to $\mathcal{Y}$, while $p(x; \theta_{\mathcal{X}})$ and $G_{\mathcal{Y}\to\mathcal{X}}$ account for the mapping of the other direction.

Specifically, as illustrated in Figure 1(1), we first sample $y_i \sim p_{\text{data}}(y)$, and then generate $\hat{x}_i = G_{\mathcal{Y}\to\mathcal{X}}(y_i; \alpha_{\mathcal{X}})$, for $i = 1, ..., n$. Starting from $\{\hat{x}_i, i = 1, ..., n\}$, we run MCMC such as Langevin dynamics for a finite number of steps toward $p(x; \theta_{\mathcal{X}})$ to obtain $\{\tilde{x}_i, i = 1, ..., n\}$, which are revised version of $\{\hat{x}_i\}$. $\{\tilde{x}_i\}$ are cooperative synthesized examples by both $p(x; \theta_{\mathcal{X}})$ and $G_{\mathcal{Y}\to\mathcal{X}}$.

$\{\tilde{x}_i\}$ are used as the synthesized examples of the energy-based model to update $\theta_{\mathcal{X}}$ according to (4). The energy-based model can teach the encoder-decoder $G_{\mathcal{Y}\to\mathcal{X}}$ via MCMC. The key is that for the cooperative synthesized examples, their sources y_i are known. In order to update $\alpha_{\mathcal{X}}$ of $G_{\mathcal{Y}\to\mathcal{X}}$, we treat $\{\tilde{x}_i, i = 1, ..., \tilde{n}\}$ as the training data for the $G_{\mathcal{Y}\to\mathcal{X}}$. Since these $\{\tilde{x}_i\}$ are obtained by the Langevin dynamics initialized from $\{\hat{x}_i\}$, which are generated by the auto-encoder with known inputs $\{y_i\}$, we can 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, i.e.,

$$L_{teach}(\alpha_{\mathcal{X}}) = \sum_{i=1}^{n} \|\tilde{x}_i - G_{\mathcal{Y}\to\mathcal{X}}(y_i, \alpha_{\mathcal{X}})\|^2. \tag{6}$$

At $\alpha_{\mathcal{X}}^{(t)}$, y_i generates 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 $\tilde{x}_i$ to chase $p(x; \theta_{\mathcal{X}})$. 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.

In the meanwhile, we simultaneously train the other pair of $p(y; \theta_{\mathcal{Y}})$ and $G_{\mathcal{X}\to\mathcal{Y}}$ in a similar way for a mapping from domain $\mathcal{X}$ to $\mathcal{Y}$. 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 align these two mapping by enforcing $G_{\mathcal{Y}\to\mathcal{X}}$ and $G_{\mathcal{X}\to\mathcal{Y}}$ to be inverse functions of each other when alternately training each pair of models via MCMC teaching. (See Figure 1(3)). This can be achieved by minimizing the following loss

$$\begin{aligned} L_{\text{inv}}(\alpha_{\mathcal{X}}, \alpha_{\mathcal{Y}}) = &\sum_{i=1}^{n} \|x_i - G_{\mathcal{Y}\to\mathcal{X}}(G_{\mathcal{X}\to\mathcal{Y}}(x_i; \alpha_{\mathcal{Y}}); \alpha_{\mathcal{X}})\|^2 \\ &+ \sum_{i=1}^{n} \|y_i - G_{\mathcal{X}\to\mathcal{Y}}(G_{\mathcal{Y}\to\mathcal{X}}(y_i; \alpha_{\mathcal{X}}); \alpha_{\mathcal{Y}})\|^2. \end{aligned} \tag{7}$$

Algorithm 1 presents a full description of the cooperative translation via alternating MCMC teaching.

4 Generalizing to unpaired cross-domain sequence translation

We can further generalize the proposed framework to learning a translation between two domains of 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 make on image sequences, we do not build distribution or mapping on sequence domain. Instead, we rely on translation model defined on image space, and bring in recurrent 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 unpaired but ordered image sequences $\{x_t, t = 1, ..., T_1\}$ and $\{y_t, t = 1, ..., T_2\}$, and $\forall x_t \in \mathcal{X}, \forall y_t \in \mathcal{Y}$. The current framework only learns translation of static image frames between two domain, without considering temporal information existing in each domain. We need to make the following two modifications to adapt it to this new task (1) We learn a recurrent 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 recurrent models for domain $\mathcal{X}$ and $\mathcal{Y}$ respectively. We learn $R_{\mathcal{X}}$ and $R_{\mathcal{Y}}$ by minimizing

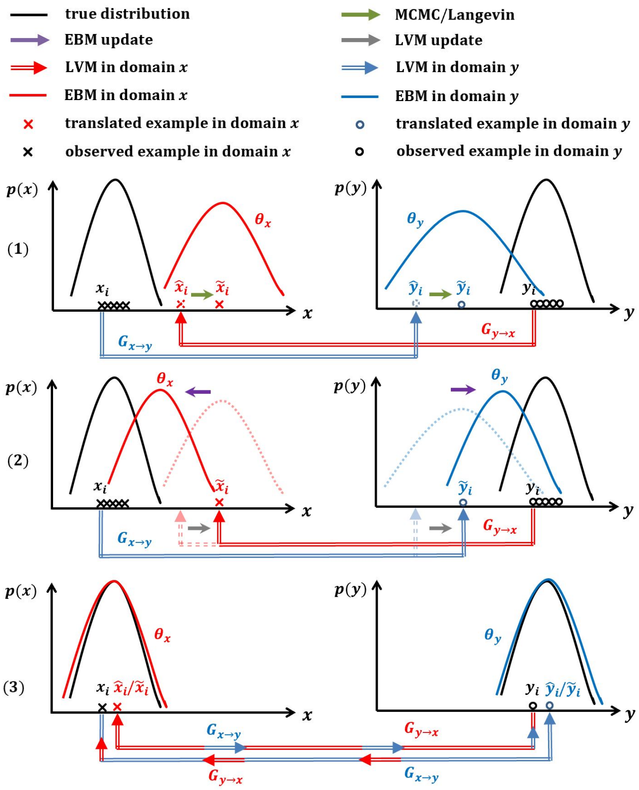


Figure 1: An illustration of alternating MCMC teaching. (1) Cooperative cross-domain translation (2) Alternating MCMC teaching of the latent variable model by the enerybased model (3) Cross-domain data alignment by enforcing mutual invertibility of the translators

$$\begin{aligned} L_{\text{rec}}(R_{\mathcal{X}}) &= \sum_t \|x_{t+k+1} - R_{\mathcal{X}}(x_{t:t+k})\|^2, \\ L_{\text{rec}}(R_{\mathcal{Y}}) &= \sum_t \|y_{t+k+1} - R_{\mathcal{Y}}(y_{t:t+k})\|^2, \end{aligned} \tag{8}$$

where $x_{t:t+k} = (x_t, ..., x_{t+k})$ and $y_{t:t+k} = (y_t, ..., y_{t+k})$. (2) With the recurrent models, we modify the loss in (7) to take into account spatial-temporal information as below

$$\begin{aligned} &L_{\text{st}}(G_{\mathcal{X}\to\mathcal{Y}}, R_{\mathcal{Y}}, G_{\mathcal{Y}\to\mathcal{X}}) \\ &= \sum_t \|x_{t+k+1} - G_{\mathcal{Y}\to\mathcal{X}}(R_{\mathcal{Y}}(G_{\mathcal{X}\to\mathcal{Y}}(x_{t:t+k})))\|^2, \\ &L_{\text{st}}(G_{\mathcal{Y}\to\mathcal{X}}, R_{\mathcal{X}}, G_{\mathcal{X}\to\mathcal{Y}}) \\ &= \sum_t \|y_{t+k+1} - G_{\mathcal{X}\to\mathcal{Y}}(R_{\mathcal{X}}(G_{\mathcal{Y}\to\mathcal{X}}(y_{t:t+k})))\|^2, \end{aligned} \tag{9}$$

Algorithm 1 Cooperative Translation

```
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) numbers 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 $t \leftarrow 0$, 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 equation (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 equation (5) to obtain $\{\tilde{y}_i\}_{i=1}^{\tilde{n}}$.
    Given $\{x\}_{i=1}^{\tilde{n}}$ and $\{\tilde{x}\}_{i=1}^{\tilde{n}}$, update $\theta_\mathcal{X}^{(t+1)} = \theta_\mathcal{X}^{(t)} + \gamma_{\theta_\mathcal{X}} \Delta(\theta_\mathcal{X}^{(t)})$, where gradient $\Delta(\theta_\mathcal{X}^{(t)})$ is (4).
    Given $\{y\}_{i=1}^{\tilde{n}}$ and $\{\tilde{y}\}_{i=1}^{\tilde{n}}$, update $\theta_\mathcal{Y}^{(t+1)} = \theta_\mathcal{Y}^{(t)} + \gamma_{\theta_\mathcal{Y}} \Delta(\theta_\mathcal{Y}^{(t)})$, where gradient $\Delta(\theta_\mathcal{Y}^{(t)})$ is (4).
    Given $\{x\}_{i=1}^{\tilde{n}}$, $\{y\}_{i=1}^{\tilde{n}}$ and $\{\tilde{x}\}_{i=1}^{\tilde{n}}$, update $\alpha_\mathcal{X}^{(t+1)} = \alpha_\mathcal{X}^{(t)} + \gamma_{\alpha_\mathcal{X}} \Delta(\alpha_\mathcal{X}^{(t)})$, where $\Delta(\alpha_\mathcal{X}^{(t)})$ is the gradient of loss functions (6) and (7).
    Given $\{x\}_{i=1}^{\tilde{n}}$, $\{y\}_{i=1}^{\tilde{n}}$ and $\{\tilde{y}\}_{i=1}^{\tilde{n}}$, update $\alpha_\mathcal{Y}^{(t+1)} = \alpha_\mathcal{Y}^{(t)} + \gamma_{\alpha_\mathcal{Y}} \Delta(\alpha_\mathcal{Y}^{(t)})$, where $\Delta(\alpha_\mathcal{Y}^{(t)})$ is the gradient of loss functions (6) and (7).
    Let $t \leftarrow t + 1$
until $t = T$
```

where $G_{\mathcal{Y}\to\mathcal{X}}(y_{t:t+k}) = (G_{\mathcal{Y}\to\mathcal{X}}(y_t), ..., G_{\mathcal{X}\to\mathcal{Y}}(x_t))$ and $G_{\mathcal{X}\to\mathcal{Y}}(x_{t:t+k}) = (G_{\mathcal{X}\to\mathcal{Y}}(x_t), ..., G_{\mathcal{Y}\to\mathcal{X}}(y_t))$ are image-wise mapping. The final objective of G and R is given by

$$\min_{G,R} L(G, R) = L_{\text{rec}}(R_\mathcal{X}) + L_{\text{rec}}(R_\mathcal{Y}) + \lambda_1 L_{\text{teach}}(G_{\mathcal{Y}\to\mathcal{X}}) + \lambda_1 L_{\text{teach}}(G_{\mathcal{X}\to\mathcal{Y}}) + \lambda_2 L_{\text{st}}(G_{\mathcal{X}\to\mathcal{Y}}, R_\mathcal{Y}, G_{\mathcal{Y}\to\mathcal{X}}) + \lambda_2 L_{\text{st}}(G_{\mathcal{Y}\to\mathcal{X}}, R_\mathcal{X}, G_{\mathcal{X}\to\mathcal{Y}}),$$

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

5 Experiments

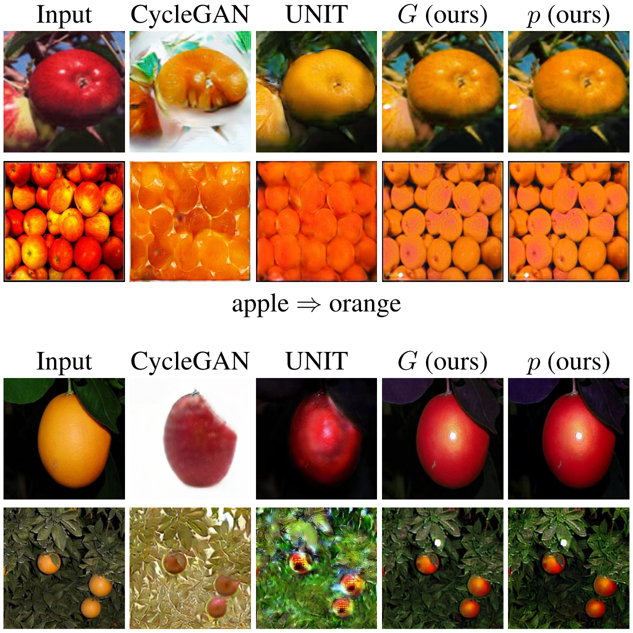
In this section, we perform experiments on the tasks of unsupervised image-to-image translation and image sequence translation to evaluate the proposed framework.

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 $p(x)$ and $p(y)$, and also the same encoder-decoder structure for $G_{\mathcal{X}\to\mathcal{Y}}$ and $G_{\mathcal{Y}\to\mathcal{X}}$.

Structure of p: The energy-based model p has a bottom-up ConvNet structure $f(Y; \theta)$ 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 100 filters. Leaky ReLU layers are applied between convolutional layers.

Structure of G. We adopt the architectures from Johnson et al. (Johnson, Alahi, and Fei-Fei 2016) for G. We use 9 residual blocks.



apple ⇒ orange

orange ⇒ apple

Figure 2: Our algorithm learns to automatically translate apples to oranges and vice versa. 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 rest columns show the translated results obtained by CycleGAN, UNIT, encoder-decoder G, and p.

Object transfiguration We train the model to translate one object category from ImageNet (Deng et al. 2009) to another. Each category has roughly 1,000 training examples. Figure 2 displays the 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 CycleGAN (Zhu et al. 2017) and UNIT (Liu, Breuel, and Kautz 2017), respectively. The last two columns show the results achieved by our model, where the fourth column displays the results generated by G without p's MCMC revision, and the fifth column displays those obtained by p's

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 baseline results of CycleGAN and UNIT. The middle two rows show the results of G and p, where s_step is the number of MCMC teaching steps. The last three rows show performances of models with different numbers of MCMC teaching steps.

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
$G(s\_step = 15)$	158.66	1.28	119.27	1.34
$p(s\_step = 15)$	**154.58**	**1.23**	**118.82**	**1.25**
$p(s\_step = 1)$	192.60	1.43	143.00	1.42
$p(s\_step = 5)$	166.41	1.43	170.38	1.40
$p(s\_step = 10)$	189.60	1.32	141.60	1.32

MCMC 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 q and p, q traces p, and p traces p_{data}, thus q and p eventually get closer and closer. The results in Table 2 verifies this fact in the sense that the results of G (fourth column) and p (fifth column) looks almost the same in appearance.

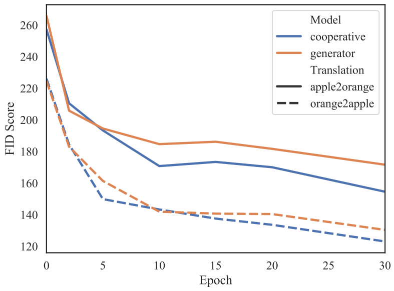
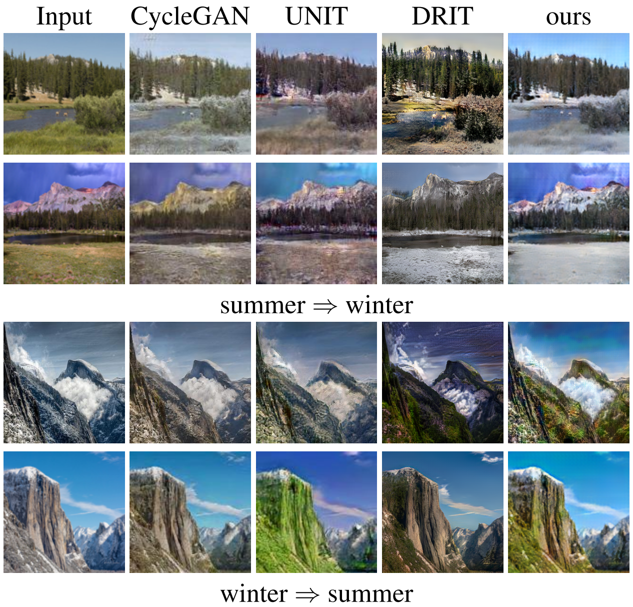


Figure 3: Analysis of FID score of the translated images (apple ⇒ orange or orange ⇒ apple) by different models over different epochs. The blue and yellow curves represent the result achieved by CoopNets system and generator respectively. The results suggest that the EBM refines the results provided by the generator.

To quantitatively evaluate the performance of the proposed model, we use the Fréchet Inception Distance (FID) (Heusel et al. 2017) for measuring 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 (Szegedy et al. 2016) model, which is pretrained on ImageNet (Deng et al. 2009) for classification purpose, as the features of each image for computing the FID. A lower FID score is desired because it means a higher similarity between the target distribution and the translated distribution.



summer ⇒ winter

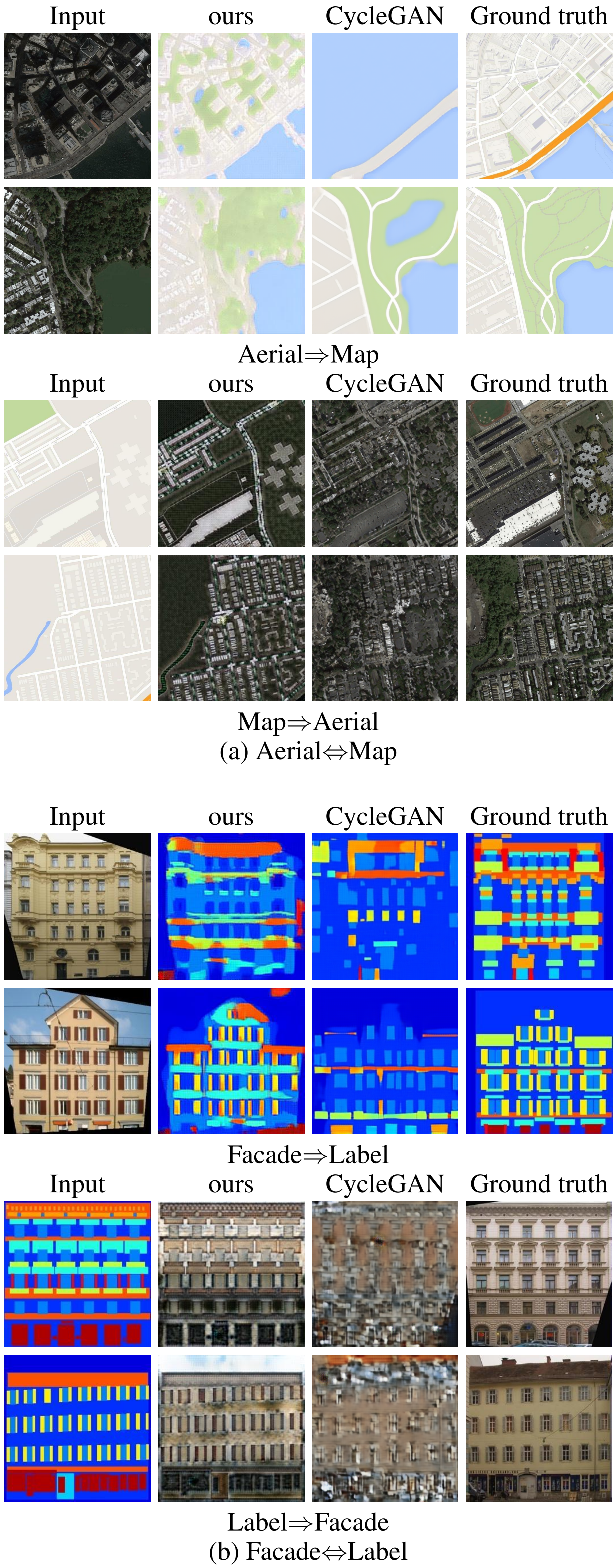
winter ⇒ summer

Figure 4: Example results of unpaired image-to-image translation 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. Our model show more realistic translation to the target domain compared with baseline methods.

We additionally evaluate our results by the domain-invariant perceptual distance (DIPD) (Huang et al. 2018), which can be used to measure the content preservation in unsupervised image-to-image translation. According to the work of (Huang et al. 2018), the DIPD is given by L2-distance between the normalized VGG (Simonyan and Zisserman 2015) 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 outperform the baseline models CycleGAN and UNIT on both metrics FID and DIPD. We also study the effects of different numbers of Langevin steps used in the model. Performances of our models with different steps of Langevin dynamics are evaluated by FID and DIPD in the last 3 rows of Table 1.

Figure 3 shows curves of FID scores of the translated images obtained by q (orange curves) and p (blue curves) over different training epochs. The solid curves present results obtained on the translation from apple to orange, while the dashed curves present results for the other direction of translation. We observe improvements in quality of the translation results in terms of FID score as the learning algorithm proceeds. We also observe the refinement effects of the MCMC revisions by the energy-based models p in the learning curves.



Aerial⇒Map

Map⇒Aerial

(a) Aerial⇔Map

Facade⇒Label

Label⇒Facade

(b) Facade⇔Label

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

Table 2: Quantitative evaluation of unsupervised image-to-image translation by PSNR on datasets Aerial⇔Map and Facade⇔Label. (adv: adversarial learning; mle: maximum likelihood estimation)

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**

Season transfer We train the model on 854 winter photos and 1,273 summer photos of Yosemite that are used in (Zhu et al. 2017) for season transfer. Figure 4 shows some qualitative results and compares against three baseline methods CycleGAN, UNIT and DRIT (Lee et al. 2020).

Translation between photo image and semantic label image We evaluate the proposed framework on two image-to-image translation datasets: (1) Aerial⇔Map (Isola et al. 2017) and (2) Facade⇔label (Tyleček and Šára 2013). These two datasets provide one-to-one paired images that are originally used for supervised image-to-image translation in (Isola et al. 2017). 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 testing stage for quantitatively evaluating unsupervised image-to-image translation models via the similarity between generated images and the corresponding ground truth. Table 2 shows a comparison of performances of our methods and some baselines, which includes CycleGAN and AlignFlow (Grover et al. 2020). AlignFlow is a generative framework that uses normalizing flows for unsupervised image-to-image translation. We consider two types of training of the AlignFlow. One is based on maximum likelihood estimation, while the other is based on adversarial learning. We measure image similarity of the translated image and the ground truth image via peak signal-to-noise ratio (PSNR), which is a suitable metric for evaluating datasets with one-to-one paring information. Figure 5 also displays some qualitative results for both datasets. Our method shows comparable results with the baseline methods.

Art style transfer We evaluate our model on collection style transfer. We learn to translate landscape photographs into art paintings in the style of Monet, Van Gogh, Cezanne and Ukiyo-e. The collections of landscape photographs are downloaded from Flickr and WikiArt and used in (Zhu et al. 2017). We train the model between photographs and each of the art collections to obtain the mapping from photographs domain to painting domain. Figure 6 displays the results.

In Figure 7, we compare our model with neural style transfer (Gatys, Ecker, and Bethge 2016b) and CycleGAN

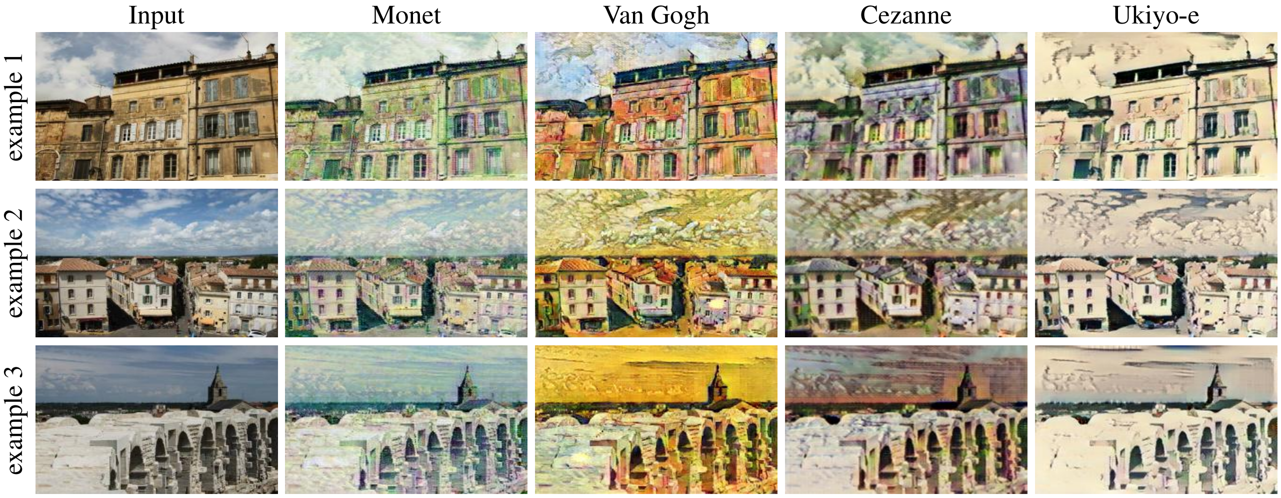


Figure 6: Collection style transfer from photo realistic images to artistic styles.

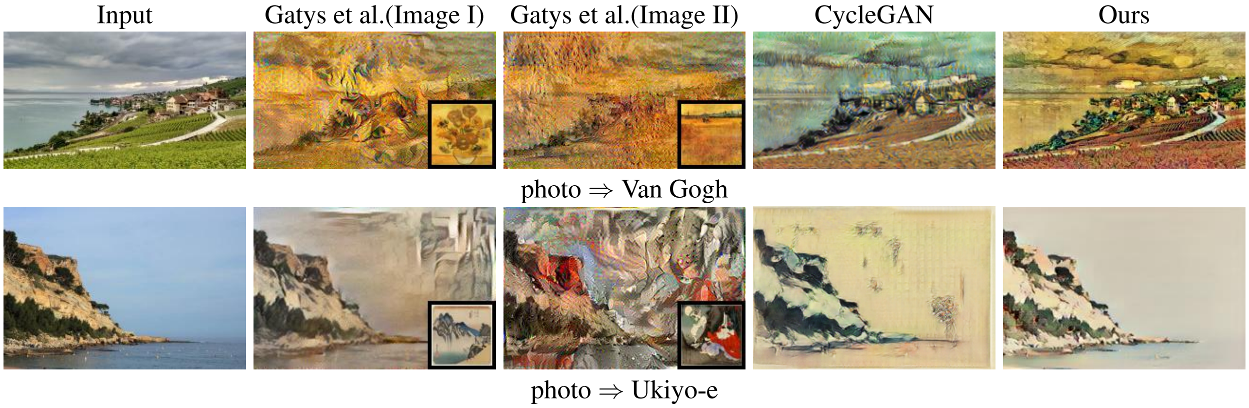


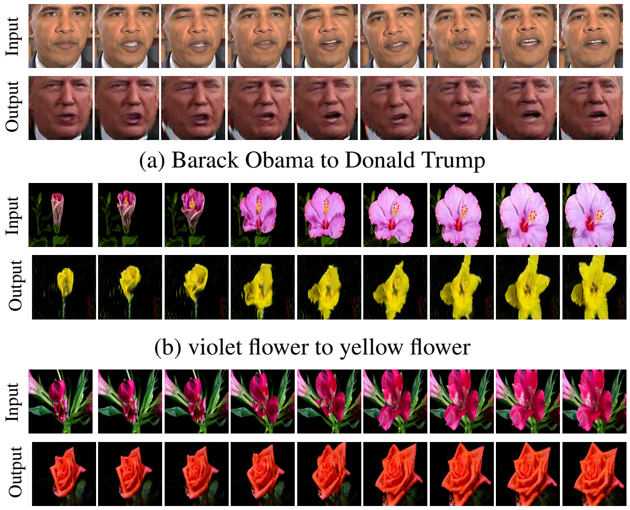
photo ⇒ Van Gogh

photo ⇒ Ukiyo-e

Figure 7: We compare our framework with style transfer method using neural network (Gatys, Ecker, and Bethge 2016a) 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 (Gatys, Ecker, and Bethge 2016a) 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.

on photo stylization. Different from our method, (Gatys, Ecker, and Bethge 2016b) 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 (Gatys, Ecker, and Bethge 2016b) 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 (Gatys, Ecker, and Bethge 2016b) is difficult to generate meaningful results, while our method succeeds to produce meaningful results that have similar styles with the target domain, which are comparable with those of CycleGAN.

Time complexity We highlight three points regarding the time complexity. (1) Although learning EBMs involves MCMC, each generator network in our framework plays a role for fast initializing the MCMC process, so that we only need a few steps of MCMC for each iteration. (2) Our MCMC method is Langevin dynamics, which is a gradient-based MCMC. It means we only need to compute the gradient of the ConvNet-parameterized energy function with respect to the image, which 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 enable an efficient offline sampling. In all, the whole proposed framework is efficient and can be scaled up for large-size datasets with current PCs and GPUs. Taking the task of style transfer on the VanGogh2photo dataset (roughly 6200 training examples) as an example, for training images of size 256×256, our training time is 0.80 seconds per iteration for



(a) Barack Obama to Donald Trump

(b) violet flower to yellow flower

(c) purple flower to red flower

Figure 8: Image sequence translation. (a) We translate Barack Obama's facial motion to Donald Trump. (b) We translate from the blooming of a violet flower to a yellow flower. (c) We translate 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.

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

In this section, we test our framework for the task of image sequence translation. We use a U-Net structure as the temporal prediction model, which takes as input a concatenation of two consecutive image frames in the past and predicts the next future frame. The U-Net structures follows the same design as the one used in (Isola et al. 2017). The energy function of each energy-based model is parameterized by a 4-layer bottom-up ConvNet structure, where the first layer has 64 5×5 filters with stride size 2, the second layer has 128 3×3 filters with stride size 2, the third layer has 256 3×3 filters with stride size 1, and the final layer is a fully connected layer with 100 filters. ReLU layers are added between convolutional layers. The step size for Langevin is 0.02. The number of Langevin dynamics steps for each energy-based model p to revise the translated examples initialized by G is 15. We adopt Adam (Kingma and Ba 2015) for optimization in our framework with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The mini-batch size is 1, and the number of parallel chains for each batch 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 (Bansal et al. 2018), in which the faces are extracted from publicly available videos of public figures, based on keypoints detected using the OpenPose library (Cao et al. 2017). The size of the image frame is 128×128 pixels. Figure 8 (b)(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 style of the input sequences and change their contents or appearances.

6 Conclusion

This paper studies unsupervised cross-domain translation problem based on a cooperative learning scheme. Our framework consist of two cooperative networks, each of which jointly trains an energy-based model and an latent variable model to account for one domain distribution. Two cooperative networks that model data distributions of two different domains are simultaneously learned and aligned by alternating MCMC teaching algorithm. Experiments show that the proposed framework can be useful for different unsupervised cross-domain translation tasks.

Acknowledgement

The work is supported by NSF DMS-2015577, DARPA XAI N66001-17-2-4029, ARO W911NF1810296, and ONR MURI N00014-16-1-2007. We thank NVIDIA for the donation of Titan Xp GPUs.

References

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

Bansal, A.; Ma, S.; Ramanan, D.; and Sheikh, Y. 2018. Recycle-GAN: Unsupervised video retargeting. In *Proceedings of the 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 *Proceedings of the 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 *Proceedings of the 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 (NeurIPS)*, 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 Nian Wu, Y. 2018. Learning generative ConvNets via multi-grid modeling and sampling. In *Proceedings of the 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 (NeurIPS)*, 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)*.

Han, T.; Lu, Y.; Zhu, S.-C.; and Wu, Y. N. 2017. Alternating Back-Propagation for generator network. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*.

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 (NeurIPS)*, 6626–6637.

Huang, X.; Liu, M.-Y.; Belongie, S.; and Kautz, J. 2018. Multimodal unsupervised image-to-image translation. In *Proceedings of the 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 *Proceedings of the 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)* 1–16.

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

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

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 *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 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*, 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 *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

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* arXiv–2004.

Xie, J.; Zheng, Z.; Fang, X.; Zhu, S.-C.; and Wu, Y. N. 2019. Multimodal Conditional Learning with Fast Thinking Policy-like Model and Slow Thinking Planner-like Model. *arXiv preprint arXiv:1902.02812* .

Xie, J.; Zheng, Z.; Gao, R.; Wang, W.; Zhu, S.-C.; and Nian Wu, Y. 2018c. Learning descriptor networks for 3D shape synthesis and analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 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 Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* .

Xie, J.; Zhu, S.-C.; and Wu, Y. N. 2017. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *Proceedings of the 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 *Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*.

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