AlignFlow: Cycle Consistent Learning
from Multiple Domains via Normalizing Flows

Aditya Grover,[*] **Christopher Chute,**[*] **Rui Shu, Zhangjie Cao, Stefano Ermon**

Department of Computer Science
Stanford University
{adityag, chute, ruishu, caozj, ermon}@cs.stanford.edu

Abstract

Given datasets from multiple domains, a key challenge is to efficiently exploit these data sources for modeling a target domain. Variants of this problem have been studied in many contexts, such as cross-domain translation and domain adaptation. We propose AlignFlow, a generative modeling framework that models each domain via a normalizing flow. The use of normalizing flows allows for a) flexibility in specifying learning objectives via adversarial training, maximum likelihood estimation, or a hybrid of the two methods; and b) learning and exact inference of a shared representation in the latent space of the generative model. We derive a uniform set of conditions under which AlignFlow is marginally-consistent for the different learning objectives. Furthermore, we show that AlignFlow guarantees exact cycle consistency in mapping datapoints from a source domain to target and back to the source domain. Empirically, AlignFlow outperforms relevant baselines on image-to-image translation and unsupervised domain adaptation and can be used to simultaneously interpolate across the various domains using the learned representation.

1 Introduction

In recent years, there has been an increase in the availability of both labeled and unlabeled datasets from multiple sources. For example, many variants of face datasets scraped from sources such as Wikipedia and IMDB are publicly available. Given data from two or more domains, we expect sample-efficient learning algorithms to be able to learn and *align* the shared structure across these domains for accurate downstream tasks. This perspective has a broad range of applications across machine learning, including relational learning (Kim et al. 2017), domain adaptation (Taigman, Polyak, and Wolf 2016; Hoffman et al. 2017; Bousmalis et al. 2017), image and video translation for computer vision (Isola et al. 2017; Wang et al. 2018), and machine translation for low resource languages (Gu et al. 2018).

Many variants of the domain alignment problem have been studied in prior work. For instance, *unpaired cross-domain translation* refers to the task of learning a mapping from one domain to another given datasets from the two domains (Zhu

et al. 2017b). This task can be used as a subproblem in the *domain adaptation* setting, where the goal is to learn a classifier for the unlabeled domain given labeled data from a related source domain (Saenko et al. 2010). Many of these problems are underconstrained due to the limitations on the labelled supervision available for target domain. An amalgam of inductive biases need to be explicitly enforced to learn meaningful solutions, e.g., cycle-consistency (Zhu et al. 2017b), entropic regularization (Courty et al. 2017) etc. These inductive biases can be specified via additional loss terms or by specifying constraints on the model family.

We present AlignFlow, a latent variable generative framework that seeks to discover the shared structure across multiple data domains using normalizing flows (Rezende and Mohamed 2015; Dinh, Krueger, and Bengio 2014; Dinh, Sohl-Dickstein, and Bengio 2017). Latent variable generative models are highly effective for inferring hidden structure within observed data from a single domain. In AlignFlow, we model the data from each domain via an invertible generative model with *a single latent space shared across all the domains*. If we let the two domains to be A and B with a shared latent space, say Z, then the latent variable generative model for A may additionally share some or all parameters with the model of domain B. Akin to a single invertible model, the collection of invertible models in AlignFlow provide great flexibility in specifying learning objectives and can be trained via maximum likelihood estimation, adversarial training or a hybrid variant accounting for both objectives.

By virtue of an invertible design, AlignFlow naturally extends as a cross-domain translation model. To translate data across two domains, say A to B, we can invert a data point from A $\rightarrow$ Z first followed by a second inversion from Z $\rightarrow$ B. Appealingly, we show that this composition of invertible mappings is *exactly cycle-consistent*, i.e., translating a datapoint from A to B using the forward mapping and backwards using the reverse mapping gives back the original datapoint and vice versa from B to A. Cycle-consistency was first introduced in CycleGAN (Zhu et al. 2017a) and has been shown to be an excellent inductive bias for underconstrained problems, such as unpaired domain alignment. While models such as CycleGAN only provide approximate cycle-consistency by incorporating additional loss

---

terms, AlignFlow can omit these terms and guarantee exact cycle-consistency by design.

We analyze the AlignFlow framework extensively. Theoretically, we derive conditions under which the AlignFlow objective is consistent in the sense of recovering the true marginal distributions. For objectives that use adversarial loss terms, we derive optimal critics in this setting. Empirically, we consider two sets of tasks: image-to-image translation and unsupervised domain adaptation. On both these tasks, we observe consistent improvements over other approximately cycle-consistent generative frameworks on three benchmark pairs of high-dimensional image datasets.

2 Preliminaries

In this section, we discuss the necessary background and notation on generative adversarial networks and normalizing flows. We overload uppercase notation X, Y, Z to denote random variables and their sample spaces and use lowercase notation x, y, z to denote values assumed by these variables.

2.1 Generative Adversarial Networks

Generative adversarial networks (GAN) are a class of latent variable generative models that specify the generator as a deterministic mapping $h : Z \rightarrow X$ between a set of latent variables Z and a set of observed variables X (Goodfellow et al. 2014). In order to sample from a GAN, we need a prior density over Z that permits efficient sampling. The generator of a GAN can also be conditional, where the conditioning is on another set of observed variables and optionally the latent variables Z as before (Mirza and Osindero 2014).

A GAN is trained via adversarial training, wherein the generator h plays a minimax game with an auxiliary critic C. The goal of the critic $C : X \rightarrow \mathbb{R}$ is to distinguish real samples from the observed dataset with samples generated via h. The generator, on the other hand, tries to generate samples that can maximally *confuse* the critic. Many learning objectives have been proposed for adversarial training, such as those based on f-divergences (Nowozin, Cseke, and Tomioka 2016) and Wasserstein Distance (Arjovsky, Chintala, and Bottou 2017). For the standard cross-entropy GAN, the critic outputs a probability of a datapoint being real and optimizes the following objective w.r.t. a data distribution $p_X^* : X \rightarrow \mathbb{R}_{\geq 0}$:

$$\mathcal{L}_{\text{GAN}}(C, h) = \mathbb{E}_{x \sim p_X^*}[\log C(x)]$$
$$+ \mathbb{E}_{z \sim p_Z}[\log(1 - C(h(z)))]. \quad (1)$$

for a suitable choice of prior density p_Z. The generator and the critic are both parameterized by deep neural networks and learned via alternating gradient updates. Because adversarial training only requires samples from the generative model, it can be used to train generative models with intractable or ill-defined likelihoods (Mohamed and Lakshminarayanan 2016). Hence, adversarial training is *likelihood-free* and in practice, it gives excellent performance for tasks that require data generation. However, these models are hard to train due to the alternating minimax optimization and suffer from issues such as mode collapse (Goodfellow 2016).

2.2 Normalizing Flows

Normalizing flows are a class of latent variable generative models that specify the generator as an *invertible* mapping $h : Z \rightarrow X$ between a set of latent variables Z and a set of observed variables X. Let p_X and p_Z denote the marginal densities defined by the model over X and Z respectively. Using the change-of-variables formula, these marginal densities can be related as:

$$p_X(x) = p_Z(z) \left| \det \frac{\partial h^{-1}}{\partial X} \right|_{X=x} \quad (2)$$

where $z = h^{-1}(x)$ due to the invertibility constraints. Here, the second term on the RHS corresponds to the absolute value of the determinant of the Jacobian of the inverse transformation and signifies the change in volume when translating across the two sample spaces.

For evaluating likelihoods via the change-of-variables formula, we require efficient and tractable evaluation of the prior density, the inverse transformation h^{-1}, and the determinant of the Jacobian of h^{-1}. To draw a sample from this model, we perform ancestral sampling, i.e., we first sample a latent vector $z \sim p_Z(z)$ and obtain the sampled vector as given by $x = h(z)$. This requires the ability to efficiently: (1) sample from the prior density and (2) evaluate the forward transformation h. Many transformations parameterized by deep neural networks that satisfy one or more of these criteria have been proposed in the recent literature on normalizing flows, e.g., NICE (Dinh, Krueger, and Bengio 2014) and Autoregressive Flows (Kingma et al. 2016; Papamakarios, Murray, and Pavlakou 2017). By suitable design of transformations, both likelihood evaluation and sampling can be performed efficiently, as in Real-NVP (Dinh, Sohl-Dickstein, and Bengio 2017). Consequently, a flow model can be trained efficiently to maximize the likelihood of the observed dataset (a.k.a. maximum likelihood estimation or MLE) as well as likelihood-free adversarial training (Grover, Dhar, and Ermon 2018).

3 The AlignFlow Framework

In this section, we present the AlignFlow framework for learning generative models in the presence of unpaired data from multiple domains. For ease of presentation, we consider the case of two domains. Unless mentioned otherwise, our results naturally extend to more than two domains as well.

3.1 Problem Setup

The learning setting we consider is as follows. We are given unpaired datasets $\mathcal{D}_A$ and $\mathcal{D}_B$ from two domains A and B respectively. We assume that the datapoints are sampled i.i.d. from some true but unknown marginal densities denoted as p_A^* and p_B^* respectively. We are interested in learning models for the following distributions: (a) the marginal likelihoods p_A and p_B that approximate p_A^* and p_B^* and (b) conditional distributions $p_{A|B}$ and $p_{B|A}$. The unconditional models can be used for density estimation and sampling from A and B whereas the conditional models can be used for translating (i.e., conditional sampling) from B $\rightarrow$ A and A $\rightarrow$ B.

Before presenting the AlignFlow framework, we note two observations. For task (a), we need datasets from the domains A and B respectively for learning. For task (b), we note that the problem is *underconstrained* since we are only given data from the marginal distributions and hence, it is unclear how to learn the conditional distribution that relates the datapoints from the two domains. Hence, we need additional inductive biases on our learning algorithm that can learn useful conditional distributions, In practice, many such forms of inductive biases have been designed and shown to be useful across relevant tasks such as cross-domain translation and domain adaptation (Zhu et al. 2017b; Liu, Breuel, and Kautz 2017).

3.2 Representation

We will use a graphical model to represent the relationships between the domains. Consider a Bayesian network A $\leftarrow$ Z $\rightarrow$ B with two sets of observed random variables (domains) A $\subseteq \mathbb{R}^n$ and B $\subseteq \mathbb{R}^n$ and a parent set of latent random variables Z $\subseteq \mathcal{Z}$.

The latent variables Z indicate a shared feature space between the observed variables A and B, which will be exploited later for efficient learning and inference. While Z is unobserved, we assume a prior density p_Z over these variables, such as an isotropic Gaussian. Finally, to compactly specify the joint distribution over all sets of variables, we constrain the relationship between A and Z, and B and Z to be invertible. That is, we specify mappings $G_{Z\rightarrow A}$ and $G_{Z\rightarrow B}$ such that the respective inverses $G_{A\rightarrow Z} = G_{Z\rightarrow A}^{-1}$ and $G_{B\rightarrow Z} = G_{Z\rightarrow B}^{-1}$ exist. Notice that such a representation naturally provides a mechanism to translate from one domain to another as the composition of two invertible mappings:

$$G_{A\rightarrow B} = G_{Z\rightarrow B} \circ G_{A\rightarrow Z} \tag{3}$$

$$G_{B\rightarrow A} = G_{Z\rightarrow A} \circ G_{B\rightarrow Z}. \tag{4}$$

Since composition of invertible mappings is invertible, both $G_{A\rightarrow B}$ and $G_{B\rightarrow A}$ are invertible. In fact, it is straightforward to observe that $G_{A\rightarrow B}$ and $G_{B\rightarrow A}$ are inverses of each other:

$$G_{A\rightarrow B}^{-1} = (G_{Z\rightarrow B} \circ G_{A\rightarrow Z})^{-1} = G_{A\rightarrow Z}^{-1} \circ G_{Z\rightarrow B}^{-1}$$
$$= G_{Z\rightarrow A} \circ G_{B\rightarrow Z} = G_{B\rightarrow A}. \tag{5}$$

3.3 Learning Algorithms & Objectives

As discussed in the preliminaries, each of the individual flows $G_{Z\rightarrow A}$ and $G_{Z\rightarrow B}$ express a model with density p_A and p_B respectively and can be trained independently via maximum likelihood estimation, adversarial learning, or a hybrid objective. However, our goal is to perform sample-efficient learning by exploiting data from other domains as well as learn a conditional mapping across the two domains. For both these goals, we require learning algorithms which use data from both domains for parameter estimation. Unless mentioned otherwise, all our results that hold for a particular domain A will have a natural counterpart for the domain B.

Adversarial Training   Instead of adversarial training of $G_{Z\rightarrow A}$ and $G_{Z\rightarrow B}$ independently, we can directly perform adversarial training of the mapping $G_{B\rightarrow A}$. That is, we first generate data from $G_{B\rightarrow A}$ using the prior density given as

p_B^*. We also introduce a critic C_A which distinguishes real samples $a \sim p_A^*$ with the generated samples $G_{B\rightarrow A}(b)$ for $b \sim p_B^*$. For example, the cross-entropy GAN loss in this case is given as:

$$\mathcal{L}_{\text{GAN}}(C_A, G_{B\rightarrow A}) = \mathbb{E}_{a\sim p_A^*}[\log C_A(a)]$$
$$+ \mathbb{E}_{b\sim p_B^*}[\log(1 - C_A(G_{B\rightarrow A}(b)))]. \tag{6}$$

The expectations above are approximated empirically via datasets $\mathcal{D}_A$ and $\mathcal{D}_B$ respectively.

Maximum Likelihood Estimation   Unlike adversarial training, flow models trained with maximum likelihood estimation (MLE) explicitly require a prior p_Z with a tractable density (e.g., isotropic Gaussian) to evaluate model likelihoods using the change-of-variables formula in Eq. 2. Due to this tractability requirement, we cannot substitute p_Z with the unknown p_B^* for MLE. Instead, we can share parameters between the two mappings. The extent of parameter sharing depends on the similarity across the two domains; for highly similar domains, entire architectures could potentially be shared in which case $G_{Z\rightarrow A} = G_{Z\rightarrow B}$.

Hybrid Training   Both MLE and adversarial training objectives can be combined into a single training objective. The most general AlignFlow objective is given as:

$$\mathcal{L}_{\text{AlignFlow}}(G_{B\rightarrow A}, C_A, C_B; \lambda_A, \lambda_B)$$
$$= \mathcal{L}_{\text{GAN}}(C_A, G_{B\rightarrow A}) + \mathcal{L}_{\text{GAN}}(C_B, G_{A\rightarrow B})$$
$$- \lambda_A \mathcal{L}_{\text{MLE}}(G_{Z\rightarrow A}) - \lambda_B \mathcal{L}_{\text{MLE}}(G_{Z\rightarrow B}) \tag{7}$$

where $\lambda_A \geq 0$ and $\lambda_B \geq 0$ are hyperparameters that control the strength of the MLE terms for domains A and B respectively. The AlignFlow objective is minimized w.r.t. the parameters of the generator $G_{A\rightarrow B}$ and maximized w.r.t. parameters of the critics C_A and C_B. Notice that $\mathcal{L}_{\text{AlignFlow}}$ is expressed as a function of the critics C_A, C_B and only $G_{B\rightarrow A}$ since the latter also encompasses the other parametric functions appearing in the objective ($G_{A\rightarrow B}, G_{Z\rightarrow A}, G_{Z\rightarrow B}$) via the invertibility constraints in Eqs. 3-5. When $\lambda_A = \lambda_B = 0$, we perform pure *adverarial training* and the prior over Z plays no role in learning. On the other hand, when $\lambda_A = \lambda_B \rightarrow \infty$, we can perform pure *MLE training* to learn the invertible generator. Here, the critics C_A, C_B play no role since the adversarial training terms are ignored.

3.4 Inference

AlignFlow can be used for both conditional and unconditional sampling at test time. For conditional sampling as in the case of domain translation, we are given a datapoint $b \in$ B and we can draw the corresponding cross-domain translation in domain A via the mapping $G_{B\rightarrow A}$. For unconditional sampling, we require $\lambda_A \neq 0$ since doing so will activate the use of the prior p_Z via the MLE terms in the learning objective. Thereafter, we can obtain samples by first drawing $z \sim p_Z$ and then applying the mapping $G_{Z\rightarrow A}$ to z. Furthermore, the same z can be mapped to domain B via $G_{Z\rightarrow B}$. Hence, we can sample paired data $(G_{Z\rightarrow A}(z), G_{Z\rightarrow B}(z))$ given $z \sim p_Z$.

4 Theoretical Analysis

The AlignFlow objective consists of three parametric models: one generator $G_{\mathrm{B}\to\mathrm{A}} \in \mathcal{G}$, and two critics $C_{\mathrm{A}} \in \mathcal{C}_{\mathrm{A}}, C_{\mathrm{B}} \in \mathcal{C}_{\mathrm{B}}$. Here, $\mathcal{G}, \mathcal{C}_{\mathrm{A}}, \mathcal{C}_{\mathrm{B}}$ denote model families specified e.g., via deep neural network based architectures. In this section, we analyze the optimal solutions to these parameterized models within well-specified model families.

4.1 Optimal Generators

Our first result characterizes the conditions under which the optimal generators exhibit *marginal-consistency* for the data distributions defined over the domains A and B.

Definition 1. *(Marginal-consistency) Let $p_{\mathrm{X,Y}}$ denote the joint distribution between two domains $\mathcal{X}$ and $\mathcal{Y}$. An invertible mapping $G_{\mathrm{Y}\to\mathrm{X}} : \mathcal{Y} \to \mathcal{X}$ is marginally-consistent w.r.t. two arbitrary distributions $(p_{\mathrm{X}}, p_{\mathrm{Y}})$ iff for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$:*

$$p_{\mathrm{X}}(x) = \begin{cases} p_{\mathrm{Y}}(y) \left| \det \frac{\partial G_{\mathrm{Y}\to\mathrm{X}}^{-1}}{\partial \mathrm{X}} \right|_{\mathrm{X}=x}, & \text{if } x = G_{\mathrm{Y}\to\mathrm{X}}(y) \\ 0, & \text{otherwise}. \end{cases}$$
(8)

Next, we show that AlignFlow is marginally-consistent for well-specified model families.

Lemma 1. *Let $\mathcal{G}_{\mathrm{A}}$ and $\mathcal{G}_{\mathrm{B}}$ denote the class of invertible mappings represented by the AlignFlow architecture for mapping $\mathrm{Z} \to \mathrm{A}$ and $\mathrm{Z} \to \mathrm{B}$. For a given choice of prior distribution p_{Z}, if there exist mappings $G_{\mathrm{Z}\to\mathrm{A}}^* \in \mathcal{G}_{\mathrm{A}}, G_{\mathrm{Z}\to\mathrm{B}}^* \in \mathcal{G}_{\mathrm{B}}$ that are marginally-consistent w.r.t. $(p_{\mathrm{A}}^*, p_{\mathrm{Z}})$ and $(p_{\mathrm{B}}^*, p_{\mathrm{Z}})$ respectively, then the mapping $G_{\mathrm{B}\to\mathrm{A}}^* = G_{\mathrm{Z}\to\mathrm{A}}^* \circ G_{\mathrm{Z}\to\mathrm{B}}^{*^{-1}}$ is marginally-consistent w.r.t. $(p_{\mathrm{A}}^*, p_{\mathrm{B}}^*)$.*

The result follows directly from Definition 1 and change-of-variables applied to the mapping $G_{\mathrm{B}\to\mathrm{A}}^*$.

Theorem 1. *Assume that the model families for the critics $C_{\mathrm{A}} : \mathrm{A} \to [0,1]$ and $C_{\mathrm{B}} : \mathrm{B} \to [0,1]$ are the set of all measurable functions for the cross-entropy GAN objective. Then, $G_{\mathrm{B}\to\mathrm{A}}^*$ (as defined in Lemma 1) globally minimizes the AlignFlow objective in Eq. 7 for any $\lambda_{\mathrm{A}} \geq 0, \lambda_{\mathrm{B}} \geq 0$.*

Proof. See Appendix A.1. Theorem 1 suggests that optimizing the AlignFlow objective will recover the marginal data distributions p_{A}^* and p_{B}^* under suitable conditions. For the other goal of learning cross-domain mappings, we note that marginally-consistent mappings w.r.t. a target data distribution (such as p_{A}^*) and a target prior density (such as p_{B}^*) need not be unique. While a cycle-consistent, invertible model family mitigates the underconstrained nature of the cross-domain translation problem, it does not provably eliminate it. We provide some non-identifiable constructions in Appendix A.3 and leave open the exploration of additional constraints that guarantee identifiability for future exploration.

4.2 Optimal Critics

Unlike standard adversarial training of an unconditional normalizing flow model (Grover, Dhar, and Ermon 2018; Danihelka et al. 2017), the AlignFlow model involves two critics. Here, we are interested in characterizing the dependence of the optimal critics for a given invertible mapping $G_{\mathrm{A}\to\mathrm{B}}$. Consider the AlignFlow framework where the GAN

(a) CycleGAN

(b) AlignFlow
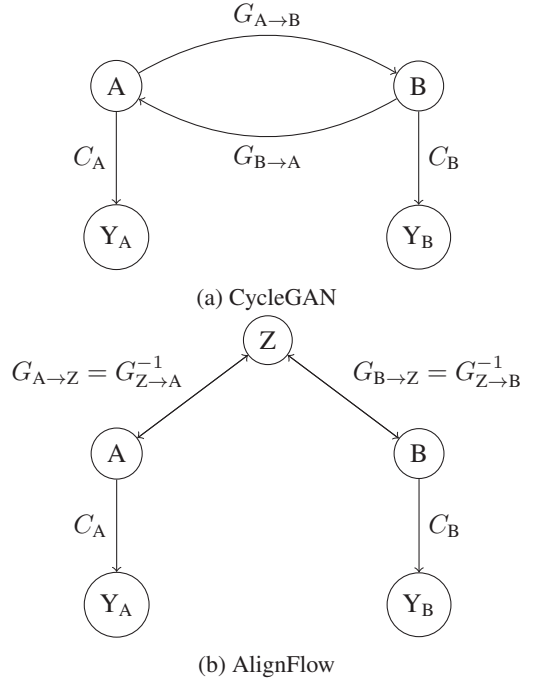
Figure 1: CycleGAN v.s. AlignFlow for unpaired cross-domain translation. Unlike CycleGAN, AlignFlow specifies a single invertible mapping $G_{\mathrm{A}\to\mathrm{Z}} \circ G_{\mathrm{B}\to\mathrm{Z}}^{-1}$ that is exactly cycle-consistent, represents a shared latent space Z between the two domains, and can be trained via both adversarial training and exact maximum likelihood estimation. Double-headed arrows denote invertible mappings. Y_{A} and Y_{B} are random variables denoting the output of the critics used for adversarial training.

loss terms in Eq. 7 are specified via the cross-entropy objective in Eq. 1. For this model, we can relate the optimal critics using the following result.

Theorem 2. *Let p_{A}^* and p_{B}^* denote the true data densities for domains A and B respectively. Let C_{A}^* and C_{B}^* denote the optimal critics for the AlignFlow objective with the cross-entropy GAN loss for any fixed choice of the invertible mapping $G_{\mathrm{A}\to\mathrm{B}}$. Letting $b = G_{\mathrm{A}\to\mathrm{B}}(a)$ for any $a \in \mathrm{A}$, we have:*

$$C_{\mathrm{A}}^*(a) = \frac{C_{\mathrm{B}}^*(b)p_{\mathrm{A}}^*(a)}{p_{\mathrm{A}}^*(a) + p_{\mathrm{B}}^*(b)(1 - C_{\mathrm{B}}^*(b)) \left| \det \frac{\partial G_{\mathrm{B}\to\mathrm{A}}^{-1}}{\partial \mathrm{A}} \right|_{\mathrm{A}=a}}.$$
(9)

Proof. See Appendix A.2. In essence, the above result shows that the optimal critic for one domain, w.l.o.g. say A, can be directly obtained via the optimal critic of another domain B for any choice of the invertible mapping $G_{\mathrm{A}\to\mathrm{B}}$, assuming access to the data marginals p_{A}^* and p_{B}^*.

4.3 Exact Cycle Consistency

So far, we have only discussed objectives that are marginally-consistent with respect to data distributions p_{A}^* and p_{B}^*. However, many domain alignment tasks such as cross-domain

translation require can be cast as learning a joint distribution $p_{A,B}^*$. As discussed previously, this problem is underconstrained given unpaired datasets $\mathcal{D}_A$ and $\mathcal{D}_B$ and the learned marginal densities alone do not guarantee learning a mapping that is useful for downstream tasks. Cycle consistency, as proposed in CycleGAN (Zhu et al. 2017a), is a highly effective learning objective that encourages learning of meaningful cross-domain mappings such that the data translated from domain A to B via $G_{A\to B}$ to be mapped back to the original datapoints in A via $G_{B\to A}$. That is, $G_{B\to A}(G_{A\to B}(a)) \approx a$ for all $a \in A$. Formally, the cycle-consistency loss for translation from A to B and back is defined as:

$$\mathcal{L}_{\text{Cycle}}(G_{B\to A}, G_{A\to B}) = E_{a\sim p_A^*}[\|G_{B\to A}(G_{A\to B}(a)) - a\|_1]. \quad (10)$$

Symmetrically, we have a cycle consistency term $\mathcal{L}_{\text{Cycle}}(G_{A\to B}, G_{B\to A})$ in the reverse direction that encourages $G_{A\to B}(G_{B\to A}(b)) \approx b$ for all $b \in B$. Next, we show that AlignFlow is *exactly* cycle consistent.

Proposition 1. *Let $\mathcal{G}$ denote the class of invertible mappings represented by an arbitrary AlignFlow architecture. For any $G_{B\to A} \in \mathcal{G}$, we have:*

$$\mathcal{L}_{\text{Cycle}}(G_{B\to A}, G_{A\to B}) = 0 \quad (11)$$

$$\mathcal{L}_{\text{Cycle}}(G_{A\to B}, G_{B\to A}) = 0 \quad (12)$$

where $G_{A\to B} = G_{B\to A}^{-1}$ by design.

The proposition follows directly from the invertible design of the AlignFlow framework (Eq. 5) and Eq. 10.

Comparison with CycleGAN We illustrate and compare AlignFlow and CycleGAN in Figure 1. CycleGAN parameterizes two independent cross-domain mappings $G_{A\to B}$ and $G_{B\to A}$, whereas AlignFlow only specifies a single, invertible mapping. Learning in a CycleGAN is restricted to an adversarial training objective along with additional cycle-consistent loss terms. In contrast, AlignFlow is exactly consistent and can be trained via adversarial learning, MLE, or a hybrid (Eq. 7) without the need for additional loss terms to enforce cycle consistency. Finally, inference in CycleGAN is restricted to conditional sampling since it does not involve any latent variables Z with easy-to-sample prior densities. As described previously, AlignFlow permits both conditional and unconditional sampling.

Comparison with UNIT and CoGAN Models such as Co-GAN (Liu and Tuzel 2016) and its extension UNIT (Liu, Breuel, and Kautz 2017) also consider adding a shared-space constraint between two different domain decoders. These models again can only enforce approximate cycle consistency and introduce additional encoder networks. Moreover, they only approximate lower bounds to the log-likelihood unlike AlignFlow which permits exact MLE training.

5 Experimental Evaluation

To achieve our two goals of data-efficient modeling of individual domains and effective cross-domain mappings, we evaluate AlignFlow on two tasks: (a) unsupervised image-to-image translation, and (b) unsupervised domain adaptation. For additional experimental details, results, and analysis beyond those stated below, we refer the reader to Appendix B.

Table 1: Mean Squared Error (MSE) comparing CycleGAN and variants of AlignFlow(AF) on paired test sets. MSE is computed pixelwise after normalizing images to $(-1, 1)$.

Dataset	Model	MSE (A $\to$ B)	MSE (B $\to$ A)
Facades	CycleGAN	0.7129	0.3286
	AF (ADV only)	0.6727	0.2679
	AF (Hybrid)	**0.5801**	**0.2512**
	AF (MLE only)	0.9014	0.5960
Maps	CycleGAN	0.0245	0.0953
	AF (ADV only)	0.0385	0.1123
	AF (Hybrid)	**0.0209**	**0.0897**
	AF (MLE only)	0.0452	0.1746
CityScapes	CycleGAN	0.1252	**0.1200**
	AF (ADV only)	0.2569	0.2196
	AF (Hybrid)	**0.1130**	0.1462
	AF (MLE only)	0.2526	0.2272

5.1 Image-To-Image Translation

We evaluate AlignFlow on three image-to-image translation datasets used by Zhu et al. (2017a): Facades, Maps, and CityScapes (Cordts et al. 2016). These datasets are chosen because they provide one-to-one aligned image pairs, so one can quantitatively evaluate unpaired image-to-image translation models via a distance metric such as mean squared error (MSE) between generated examples and the corresponding ground truth. While MSE can be substituted for perceptual losses in other scenarios, it is a suitable metric for evaluating datasets with one-to-one ground pairings. Note that the task at hand is unpaired translation and hence, the pairing information is omitted during training and only used for evaluation.

We report the MSE for translations on the test sets after cross-validation of hyperparameters in Table 1. For hybrid models, we set $\lambda_A = \lambda_B$ and report results for the best values of these hyperparameters. We observe that while learning AlignFlow via adversarial training or MLE alone is not as competitive as CycleGAN, hybrid training of AlignFlow significantly outperforms CycleGAN in almost all cases. Specifically, we observe that MLE alone typically performs worse than adversarial training, but together both these objectives seem to have a regularizing effect on each other. Qualitative interpolations on the Facades dataset are shown in Figure 2.

5.2 Unsupervised Domain Adaptation

In unsupervised domain adaptation (Saenko et al. 2010), we are given data from two related domains: a source and a target domain. For the source, we have access to both the input datapoints and their labels. For the target, we are only provided with input datapoints without any labels. Using the available data, the goal is to learn a classifier for the target domain. We extend Hoffman et al. (2017) to use an AlignFlow architecture and objective (adversarially trained Real-NVPs (Dinh, Sohl-Dickstein, and Bengio 2017) here) in place of CycleGAN for this task.

A variety of algorithms have been proposed for the above task which seek to match pixel-level or feature-level distributions across the two domains. See Appendix B for more
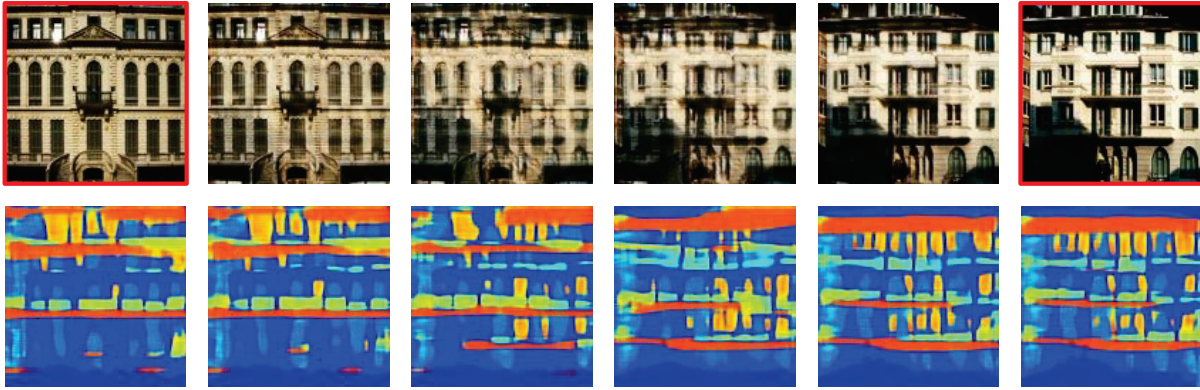
Figure 2: Latent space interpolation on Facades. **Top:** Left- and right-most images are sampled from $\mathcal{D}_A$ (red boxes). Interpolation is then performed in latent space and then decoded using $G_{Z \to A}$. We see semantically meaningful changes across the row, *e.g.,* in the shadow and the style of entrance to the building. **Bottom:** For each image in the top row, its latent representation is decoded into the target domain using $G_{Z \to B}$. Inspection of the orange regions indicates a change from 3 floors (left) to 4 floors (right).

Table 2: Test classification accuracies for domain adaptation from source → target. The **source only** and **target only** models directly use classifiers trained on the source and target datasets respectively. Baseline numbers taken from the cited works.

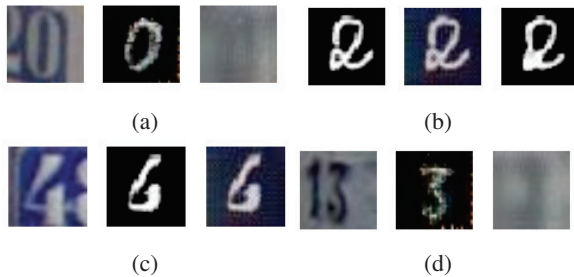| Model	MNIST → USPS	USPS → MNIST	SVHN → MNIST
source only	82.2 ± 0.8	69.6 ± 3.8	67.1 ± 0.6
ADDA (Tzeng et al. 2017)	89.4 ± 0.2	90.1 ± 0.8	76.0 ± 1.8
CyCADA (Hoffman et al. 2017)	95.6 ± 0.2	96.5 ± 0.1	90.4 ± 0.4
UNIT (Liu, Breuel, and Kautz 2017)	95.97	93.58	90.53
AlignFlow	$\mathbf{96.2 \pm 0.2}$	$\mathbf{96.7 \pm 0.1}$	$\mathbf{91.0 \pm 0.3}$
target only	96.3 ± 0.1	99.2 ± 0.1	99.2 ± 0.1

(a)   (b)

(c)   (d)

Figure 3: Examples of failure modes for CycleGAN reconstructions in SVHN ↔ MNIST cross-domain translation. In each group of three images in (a-d), a real example from the source domain (SVHN) is shown on the left, the translated image in the target domain (MNIST) is shown at center, and the reconstructed image in the source domain based on the translation is shown on the right.

details. For fair and relevant comparison, we compare against baselines Cycada (Hoffman et al. 2017) and UNIT (Liu, Breuel, and Kautz 2017) which involve pixel-level translations and are closest to the current work. We evaluate across all pairs of source and target datasets as in Hoffman et al. (2017) and Liu, Breuel, and Kautz (2017): MNIST, USPS, SVHN, which are all image datasets of handwritten digits

with 10 classes. In Table 2, we see that AlignFlow outperforms both Cycada (Hoffman et al. 2017) (based on Cycle-GAN) and UNIT (Liu, Breuel, and Kautz 2017) in all cases. Combining AlignFlow with other state-of-the-art adaptation approaches e.g., Shu et al. (2018), Long et al. (2018), Kumar et al. (2018), Liu et al. (2018a), Sankaranarayanan et al. (2018), Liu et al. (2018b) is an interesting direction for future work.

In Figure 3, we show some failure modes of using approximately cycle-consistent objectives for the Cycada model. Notice that the image label and style changes or becomes unrecognizable in translating and reconstructing the input. In contrast, AlignFlow is exactly cycle consistent and hence, the source reconstructions based on the translated images will be exactly the source image by design.

5.3 Multi-Domain Concurrent Interpolations

The use of a shared latent space in AlignFlow allows us to perform paired interpolations in two domains simultaneously. While pure MLE without any parameter sharing does not give good alignment, pure adversarial training cannot be used for unconditional sampling since the prior p_Z is inactive. Hence, we use AlignFlow models trained via a hybrid objective for latent space interpolations. In particular, we sample two datapoints $a', a'' \in \mathcal{D}_A$ and obtain their latent representations $z', z'' \in \mathcal{Z}$ via $G_{Z \to A}$. Following Dinh, Sohl-Dickstein, and
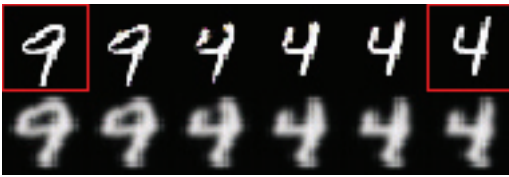
(a) MNIST→USPS

(b) USPS→MNIST

Figure 4: Multi-domain latent space interpolations. **Top:** Left-most and right-most images are sampled from $\mathcal{D}_\text{A}$ (in red boxes). Interpolation is then performed in latent space and then decoded using $G_{\text{Z}\rightarrow\text{A}}$. **Bottom:** For each corresponding image in the top row, its latent representation is decoded into the target domain using $G_{\text{Z}\rightarrow\text{B}}$. Note how both class identity and style are preserved in the interpolated pairs of digits in the two domains. Also, notice that the USPS images (even the true ones in red boxes) are slightly blurred due to the upscaling applied as standard preprocessing.

Bengio (2017), we compute interpolations in the polar space as $\tilde{z} = z' \sin\phi + z'' \cos\phi$ for several values of $\phi \in (0, 2\pi)$. Finally, we map $\tilde{z}$ to either back to domain A via $G_{\text{Z}\rightarrow\text{A}}$ and B via $G_{\text{Z}\rightarrow\text{B}}$. We show this empirically on the MNIST/USPS datasets in Figure 4. We see that many aspects of style and content are preserved in the interpolated samples.

6 Related Work

A key assumption in unsupervised domain alignment is the existence of a deterministic or stochastic mapping $G_{\text{A}\rightarrow\text{B}}$ such that the distribution of B matches that of $G_{\text{A}\rightarrow\text{B}}(\text{A})$, and vice versa. This assumption can be incorporated as a marginal distribution-matching constraint into the objective using an adversarially-trained GAN critic (Goodfellow et al. 2014). However, this objective is under-constrained. To partially mitigate this issue, CycleGAN (Zhu et al. 2017a), DiscoGAN (Kim et al. 2017), and DualGAN (Yi et al. 2017) added an approximate cycle-consistency constraint that encourages $G_{\text{B}\rightarrow\text{A}} \circ G_{\text{A}\rightarrow\text{B}}$ and $G_{\text{A}\rightarrow\text{B}} \circ G_{\text{B}\rightarrow\text{A}}$ to behave like identity functions on domains A and B respectively. While cycle-consistency is empirically very effective, alternatives based on variational autoencoders that do not require either cycles or adversarial training have also been proposed recently (Hoshen 2018; Hoshen and Wolf 2018).

Models such as CoGAN (Liu and Tuzel 2016), UNIT (Liu, Breuel, and Kautz 2017), and CycleGAN (Zhu et al. 2017a) have since been extended to enable one-to-many mappings (Huang et al. 2018b; Zhu et al. 2017b) as well as multi-domain alignment (Choi et al. 2018). Our work focuses on the one-to-one unsupervised domain alignment setting. In contrast to previous models, AlignFlow leverages both a shared latent space and *exact* cycle-consistency. To our knowledge, AlignFlow provides the first demonstration that invertible models can be used successfully in lieu of the cycle-consistency objective. Furthermore, AlignFlow allows the incorporation of exact maximum likelihood training, which we demonstrated to induce a meaningful shared latent space that is amenable to interpolation.

7 Conclusion & Future Work

We presented AlignFlow, a generative framework for learning from multiple data sources based on normalizing flow models. AlignFlow has several attractive properties: it guarantees exact cycle-consistency via a single cross-domain mapping, learns a shared latent space across two domains, and permits a flexible training objective which can combine adversarial training and exact maximum likelihood estimation. Theoretically, we derived conditions under which the AlignFlow model learns marginals that are consistent with the underlying data distributions. Finally, our empirical evaluation demonstrated significant gains on unsupervised domain translation and adaptation, and an increase in inference capabilities, e.g., paired interpolations in the latent space for two domains.

In the future, we plan to consider extensions of Align-Flow for learning stochastic, multimodal mappings (Zhu et al. 2017b) and translations across more than two domains (Choi et al. 2018). Exploring recent advancements in invertible architectures e.g., Ho et al. (2019), Huang et al. (2018a), Chen et al. (2018), Grathwohl et al. (2018) within AlignFlow is another natural promising direction for future work. With a handle on model likelihoods and invertible inference, we are optimistic that AlignFlow can aid the characterization of useful structure that guarantees identifiability in underconstrained problems such as domain alignment (Cui et al. 2014; Galanti, Wolf, and Benaim 2017; Alvarez-Melis, Jegelka, and Jaakkola 2019; Wu et al. 2019).

References

Alvarez-Melis, D.; Jegelka, S.; and Jaakkola, T. S. 2019. Towards optimal transport with global invariances. In *AISTATS*.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. In *ICML*.

Bousmalis, K.; Silberman, N.; Dohan, D.; Erhan, D.; and Krishnan, D. 2017. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*.

Chen, T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. In *NeurIPS*.

Choi, Y.; Choi, M.; Kim, M.; Ha, J.-W.; Kim, S.; and Choo, J. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *CVPR*.

Courty, N.; Flamary, R.; Habrard, A.; and Rakotomamonjy, A. 2017. Joint distribution optimal transportation for domain adaptation. In *NeurIPS*.

Cui, Z.; Chang, H.; Shan, S.; and Chen, X. 2014. Generalized unsupervised manifold alignment. In *NeurIPS*.

Danihelka, I.; Lakshminarayanan, B.; Uria, B.; Wierstra, D.; and Dayan, P. 2017. Comparison of maximum likelihood and gan-based training of real nvps. *arXiv preprint arXiv:1705.05263*.

Dinh, L.; Krueger, D.; and Bengio, Y. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2017. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Galanti, T.; Wolf, L.; and Benaim, S. 2017. The role of minimal complexity functions in unsupervised learning of semantic mappings. *arXiv preprint arXiv:1709.00074*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*.

Goodfellow, I. 2016. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

Grathwohl, W.; Chen, R. T.; Betterncourt, J.; Sutskever, I.; and Duvenaud, D. 2018. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*.

Grover, A.; Dhar, M.; and Ermon, S. 2018. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *AAAI Conference on Artificial Intelligence*.

Gu, J.; Hassan, H.; Devlin, J.; and Li, V. O. 2018. Universal neural machine translation for extremely low resource languages. *arXiv preprint arXiv:1802.05368*.

Ho, J.; Chen, X.; Srinivas, A.; Duan, Y.; and Abbeel, P. 2019. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *ICML*.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A. A.; and Darrell, T. 2017. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*.

Hoshen, Y., and Wolf, L. 2018. Nam: Non-adversarial unsupervised domain mapping. In *ECCV*.

Hoshen, Y. 2018. Non-adversarial mapping with vaes. In *NeurIPS*.

Huang, C.-W.; Krueger, D.; Lacoste, A.; and Courville, A. 2018a. Neural autoregressive flows. *arXiv preprint arXiv:1804.00779*.

Huang, X.; Liu, M.-Y.; Belongie, S.; and Kautz, J. 2018b. Multimodal unsupervised image-to-image translation. In *ECCV*.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *CVPR*.

Kim, T.; Cha, M.; Kim, H.; Lee, J. K.; and Kim, J. 2017. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*.

Kingma, D. P.; Salimans, T.; Jozefowicz, R.; Chen, X.; Sutskever, I.; and Welling, M. 2016. Improved variational inference with inverse autoregressive flow. In *NeurIPS*.

Kumar, A.; Sattigeri, P.; Wadhawan, K.; Karlinsky, L.; Feris, R.; Freeman, B.; and Wornell, G. 2018. Co-regularized alignment for unsupervised domain adaptation. In *NeurIPS*.

Liu, M.-Y., and Tuzel, O. 2016. Coupled generative adversarial networks. In *NeurIPS*.

Liu, A. H.; Liu, Y.-C.; Yeh, Y.-Y.; and Wang, Y.-C. F. 2018a. A unified feature disentangler for multi-domain image translation and manipulation. In *NeurIPS*.

Liu, Y.-C.; Yeh, Y.-Y.; Fu, T.-C.; Wang, S.-D.; Chiu, W.-C.; and Frank Wang, Y.-C. 2018b. Detach and adapt: Learning cross-domain disentangled deep representation. In *CVPR*.

Liu, M.-Y.; Breuel, T.; and Kautz, J. 2017. Unsupervised image-to-image translation networks. In *NeurIPS*.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional adversarial domain adaptation. In *NeurIPS*.

Mirza, M., and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Mohamed, S., and Lakshminarayanan, B. 2016. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*.

Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. f-gan: Training generative neural samplers using variational divergence minimization. In *NeurIPS*.

Papamakarios, G.; Murray, I.; and Pavlakou, T. 2017. Masked autoregressive flow for density estimation. In *NeurIPS*.

Rezende, D. J., and Mohamed, S. 2015. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *ECCV*.

Sankaranarayanan, S.; Balaji, Y.; Castillo, C. D.; and Chellappa, R. 2018. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*.

Shu, R.; Bui, H. H.; Narui, H.; and Ermon, S. 2018. A dirt-t approach to unsupervised domain adaptation. In *ICLR*.

Taigman, Y.; Polyak, A.; and Wolf, L. 2016. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *CVPR*.

Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Liu, G.; Tao, A.; Kautz, J.; and Catanzaro, B. 2018. Video-to-video synthesis. In *NeurIPS*.

Wu, Y.; Winston, E.; Kaushik, D.; and Lipton, Z. 2019. Domain adaptation with asymmetrically-relaxed distribution alignment. *arXiv preprint arXiv:1903.01689*.

Yi, Z.; Zhang, H.; Tan, P.; and Gong, M. 2017. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017a. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.

Zhu, J.-Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A. A.; Wang, O.; and Shechtman, E. 2017b. Toward multimodal image-to-image translation. In *NeurIPS*.