2022

Manipulating natural images by learning relationships between visual domains

BOSTON UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

Dissertation

MANIPULATING NATURAL IMAGES BY LEARNING RELATIONSHIPS BETWEEN VISUAL DOMAINS

by

BEN USMAN

B.S., Moscow Institute of Physics and Technology, 2014

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2022

Approved by

First Reader

Kate Saenko, PhD
Associate Professor of Computer Science

Second Reader

Brian Kulis, PhD
Associate Professor of Computer Science

Third Reader

Bryan Plummer, PhD
Assistant Professor of Computer Science

Fourth Reader

Sarah Adel Bargal, PhD
Research Assistant Professor of Computer Science

Acknowledgments

First and foremost, I would like to thank my amazing advisor, Kate Saenko, who accepted me into her group, let me follow my curiosity for six long years, and showed me how to turn my passion for research into an actual profession, which ultimately, completely and irreversibly changed my life for better. With superhuman patience, she taught me how to distill my curiosity into verifiable research ideas, and how to communicate these ideas to the world, for which I am forever grateful.

I would also like to thank my wife (and the best co-author I ever had) Dina, my parents Aleksandr and Iulia, my sister Meri, and my grandmother Irina, who were supportive of my “long journey into the unknown”, throughout all these years full of all up and downs, long and exhausting struggles, and short but exhilarating victories.

I would like to thank the Computer Science department of Boston University, for giving me an opportunity to learn from amazing people there, and to Skolkovo Institute of Technology for giving me an opportunity to do research at many different places that lead me to my first timid steps towards the scientific career, and my mentor there, Ivan Oseledets, for unconditionally supporting my research interests, and helping me with these first timid steps. I would like to thank the Moscow Institute of Physics and Technology, and specifically, the Department of Innovation and High Technology that introduced me to the amazing world of computer science and like-minded peers - the first place where I truly felt that I belonged. I would also like to thank teachers that throughout the years slowly helped me to gain the confidence I needed to make this leap of faith, including Prof. Andrey Gavrikov, Prof. Grigory Ivanov, and my late high school physics teacher, Beloshapskaya Kira Aleksandrovna, who was among the first to see potential in me and encourage my interest in science.

MANIPULATING NATURAL IMAGES BY LEARNING RELATIONSHIPS BETWEEN VISUAL DOMAINS

BEN USMAN

Boston University, Graduate School of Arts and Sciences, 2022

Major Professor: Kate Saenko, PhD
Associate Professor of Computer Science

ABSTRACT

Manipulation of visual attributes of real images is a fundamental generative computer vision task. The goal is to alter specified visual attributes of a given input image while preserving all other visual attributes. The manipulations can be global, such as changes in lighting or view angle, or spatially localized, such as the addition or removal of individual objects or actors, changes to their appearance, pose, or expression. The majority of existing attribute manipulation methods are either hand-crafted for a very specific manipulation (e.g. Photoshop filters) or require a large dataset with attribute annotations to learn the desired manipulation in a supervised fashion. This requirement renders fully-supervised methods prohibitively expensive to apply in many real application domains that do not have large densely annotated datasets. In this thesis, we investigate whether flexible attribute manipulation models can be trained without massive labeled datasets of real images by transferring knowledge about the desired manipulation across different image datasets (domains) that share the underlying structure. This transfer is often performed by transforming examples from one domain in a way that makes them indistinguishable from the other for a given family of neural discriminators. This procedure is called unsupervised adversarial image alignment, and in this thesis we show that it suffers from training instability, and introduce two new approaches for the stabilization of this alignment: objective dualization and likelihood-ratio minimizing flows. After that, we propose a novel setup and a method for manipulation of natural images that uses only cross-domain supervision. Finally, we propose a new method for the manipulation of domain-specific and domain-invariant factors of variation in the absence of any supervision in either domain. We show that the proposed cross-domain alignment objectives yield more stable solutions and that the proposed cross-domain image manipulation techniques successfully learn correspondences between factors of variation present across different visual domains.

Contents

List of Tables

List of Figures

List of Abbreviations

300-VW	300 Videos in the Wild
3DMM	3D Morphable Models
ADDA	Adversarial Discriminative Domain Adaptation
AoI	Attribute of Interest
CGI	Computer-Generated Imagery
CNN	Convolutional Neural Network
CORAL	CORrelation ALignment
DA	Domain Adaptation
DAN	Domain Adaptation Networks
DANN	Domain-Adversarial Neural Networks
EMD	Earth Mover's Distance
FFJORD	Free-form Jacobian of Reversible Dynamics
GAN	Generative Adversarial Networks
GAN	Generative Adversarial Network
GLOW	Generative Flow
GPU	Graphics Processing Unit
GT	Ground Truth
JAN	Joint Adaptation Networks
JSD	Jensen Shannon Divergence
k-NN	k-Nearest Neighbors
KL	Kullback–Leibler divergence
LRMF	Likelihood Ratio Minimizing Flows
LSTM	Long Short-Term Memory
MI	Mutial Information
MMD	Maximum Mean Discrepancy
MNIST	Modified National Institute of Standards and Technology
Real	NVP Real-valued Non-Volume Preserving
RIFT	Restricted Information Flow for Translation
RKHS	Reproducing Kernel Hilbert Space
SVHN	Street View House Numbers database
UMMI2I	Unsupervised Many-to-Many Image-to-Image
UNIT	UNsupervised Image-to-image Translation Networks
WGAN	Wasserstein Generative Adversarial Network

Chapter 1

Introduction

Ever since the first photograph was taken by Joseph Nicephore Niepce with the aid of the camera obscura in the beginning of the 19th century [96], the art of realistic manipulation of captured imagery has been continuously developing. An early example of such manipulation, shown in Figure 1·1, depicts a military general on a horse in front of a battlefield. All parts of this image were sourced from different pre-existing images by stacking multiple negative strips. In the decades following early experiments of Kirsch et al. [58], digital image capture flourished, and eventually penetrated all spheres of modern life. Early digital image manipulation software [108] mimicked classical forgery techniques, providing tools to manually split captured images into “layers”, and stack these layers across multiple images, similar to how authors of the fake military general photograph stacked photographic film. While powerful, these tools require a lot of skill and can not be applied at scale if one needs to perform a manipulation consistently across a large collection of images, *e.g.* for video manipulation. In an attempt to streamline face manipulation across video frames, computer-generated (CGI) face manipulation techniques used in movie production [12] often relied on *parametric* morphable face models that explicitly model face shape, lighting, albedo, reflectance, etc. As shown in Figure 1·2, these approaches reduce image manipulation into three simpler sub-tasks: approximating the input image by fitting the parameters of the parametric model to it, manipulating inferred parameter values as needed to perform the desired manipulation, and rendering an

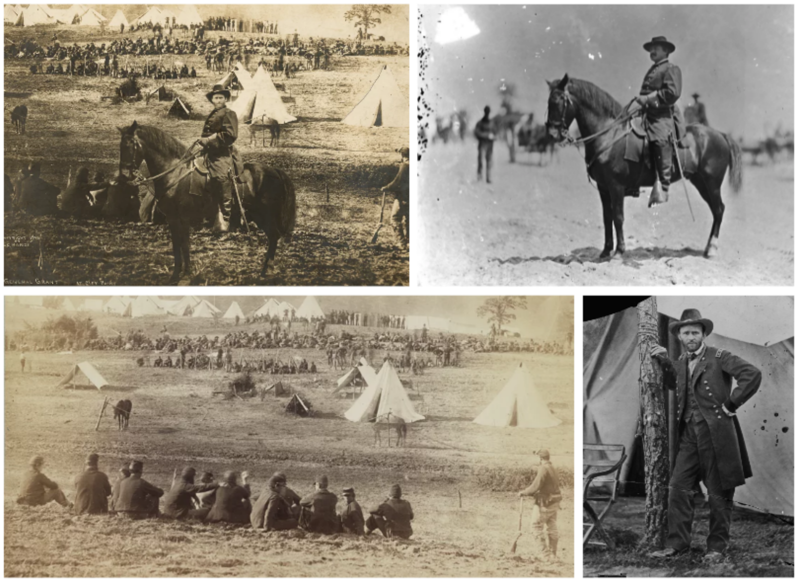


Figure 1·1: An example of image forgery (1902). The final image (top left) was created artificially by combining a horseman, a battlefield, and a face from three other photographs. *General Grant at City Point, Library of Congress* [23].

image back from the manipulated parameter vector. Unfortunately, building accurate and visually plausible parametric domain models even for a relatively simple problem domain, such as human faces, turned out to be an extremely challenging problem that took decades of research to get even remotely close to realism. And while more recent learning-based approaches that use parametric face models show a lot of potential [28, 114], we argue that the sheer amount of intellectual effort required for building such hand-crafted parametric models for each application domain from scratch renders them poor candidates for building generic image manipulation tools in the future.

Beyond addition and removal of objects and manipulation of facial expression, practitioners might be interested in being able to manipulate arbitrary atomic and uniquely identifiable properties of images - that we further refer to as “image attributes”. Recently, fully-supervised *neural* methods [19, 20, 66] were shown to be

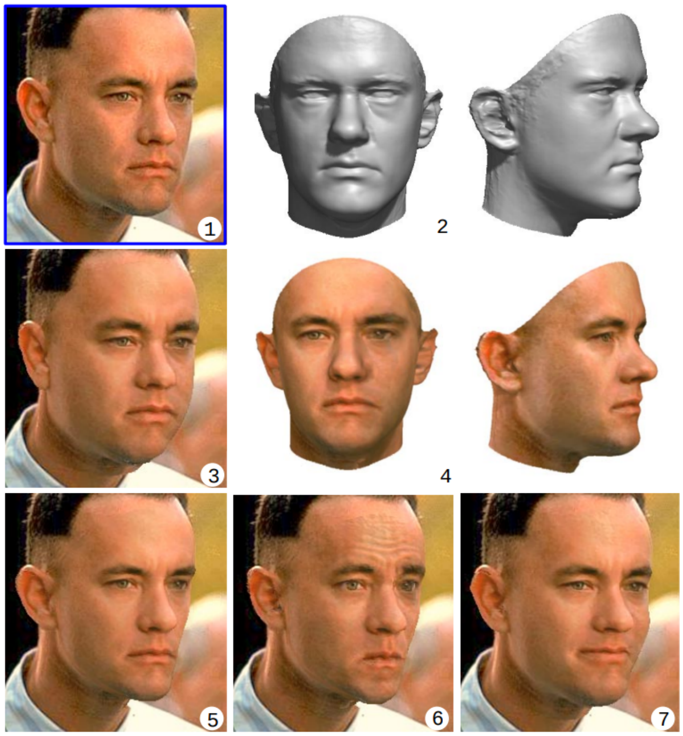


Figure 1·2: Face manipulation using morphable face models [12] (1999) factors the problem into three sub-problems: fitting a 3D mesh and texture to the face image, manipulating these meshes and textures, and rendering them back into the image domain.

able to learn how to manipulate specified attributes of real images while preserving other attributes using explicit attribute labels, as demonstrated in Figure 1·3. These methods require dense attribute annotations for each real image, which are often prohibitively expensive to acquire. Moreover, some visual attributes, like reflectance or lighting maps, can not be easily labeled by humans at all. To sum up, most image manipulation approaches extensively explored throughout the last two decades require either a lot of time and skill to manually manipulate each image, decades of research to build accurate parametric models of the application domain, or large manually annotated datasets that are either expensive or impossible to acquire for many real-world application domains. In this thesis, we explore an alternative direc-

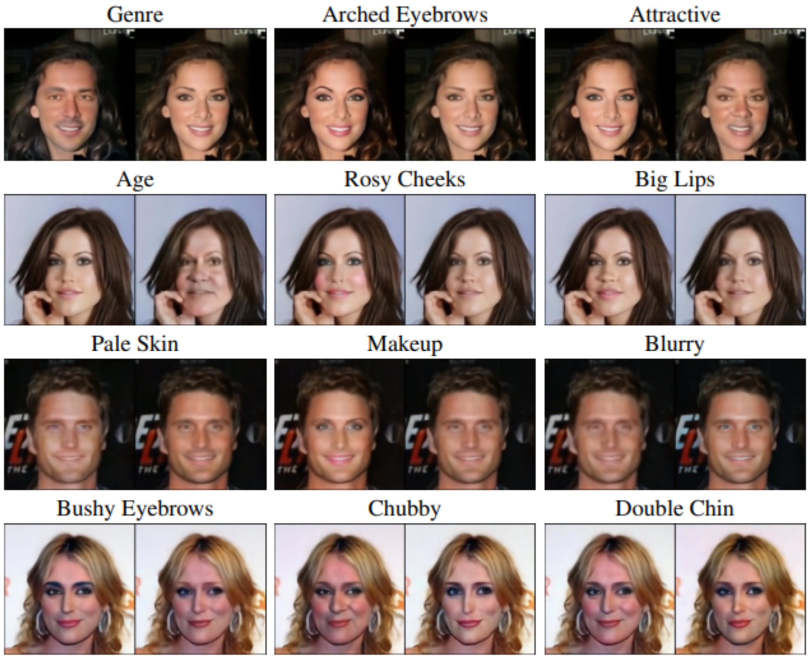


Figure 1·3: Manipulation of natural images using a fully-supervised neural method [66] requires a large annotated dataset for supervision. Such densely annotated datasets are prohibitively expensive to acquire in many real-world application domains.

tion: building methods that manipulate real images by inferring relationships between factors of variability present across structurally similar but visually distinct datasets, that we further refer to as "different visual domains". Some examples of such visually distinct but structurally similar domains, provided in Figure 1·4, include datasets of winter-time and summer-time photos, horses and zebras, etc. In Section 4.1, we show that we can manipulate various aspects of real images using a crude simulation without relying on any additional supervision by inferring relationships between factors of variability present across real and synthetic domains.

First, we note that in order to learn structural relationships between arbitrary domain pairs that, unlike 3D morphable models (3DMMs), lack out-of-the-box image fitting capabilities, we need a universal mechanism for discovering and relating visual

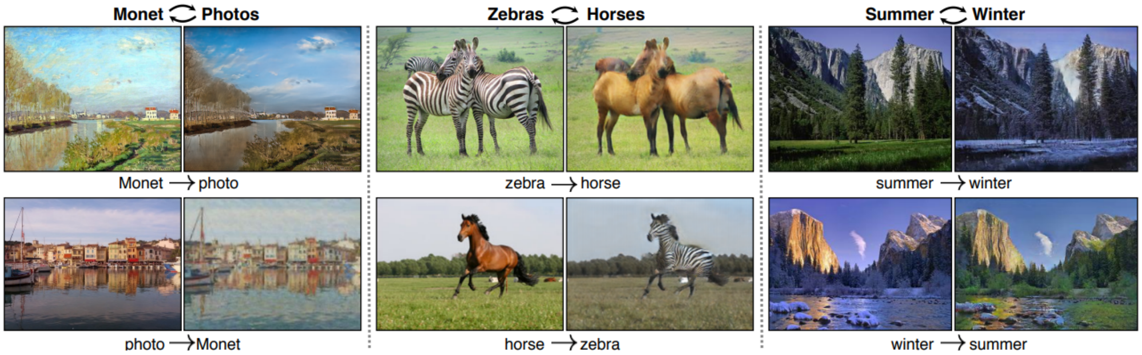


Figure 1·4: Examples of visual domains. Landscape paintings and landscape photos, images of horses and zebras, winter-time and summer-time photos - these dataset pairs are visually distinct but structurally similar (image from the work of Zhu et al. [127]).

domains that do not assume access to pairs of corresponding images from these domains during training. Being able to infer a mapping that turns images from “source” domain into plausible examples from the ”target” domain using unpaired examples of images from two domains is the first step towards reaching our ultimate goal of learning meaningful structural relationships between visual domains. This problem is called “unsupervised adversarial alignment”, and prior work addressing it [21, 71] focuses on finding a transformation that minimizes some notion of “distinguishability” between real examples of images from the target domain and transformed source images. In this thesis, we show that existing notions of distinguishability lack either expressivity or stability and semantic consistency. As a result, methods capable of producing high-quality alignment in higher dimensional spaces, such as spaces of images, are often prone to training instability or to producing semantically nonsensical solutions. To address these limitations, we propose several techniques for improving the stability and semantic consistency of unsupervised adversarial alignment without sacrificing the expressive power of trained models using objective dualization [50], normalizing flows [98], and adversarial defense techniques [112].

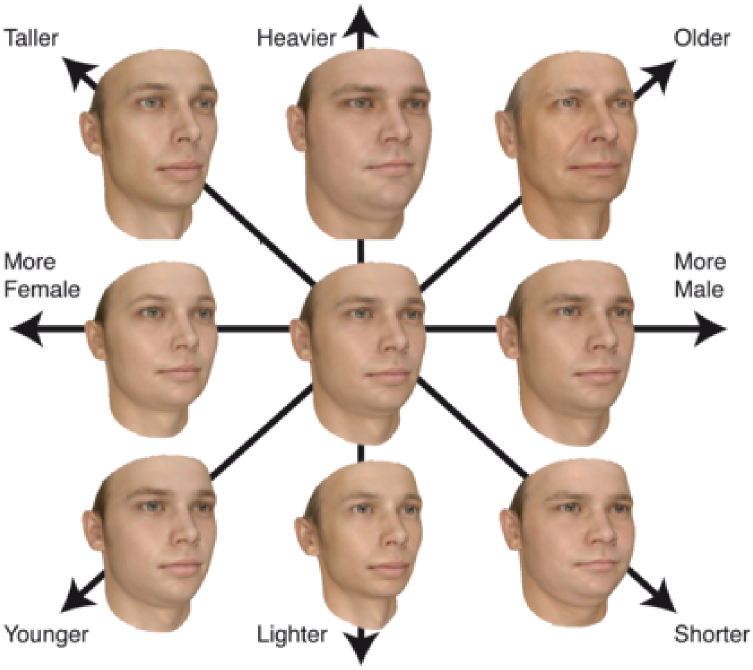


Figure 1·5: 3D Morphable Basel Face Model [66] does not model many aspects of in-the-wild faces, such as hair or shadows cast by other objects in the scene, but it still can be used to learn about meaningful degrees of variability in real data. In this thesis, we focus on relating such degrees of variability across different visual domain pairs.

Next, we note that a simulation that shares the underlying structural degrees of variability with the application domain can provide a lot of useful learning signals about the structure of that problem domain, even if its visual fidelity and completeness are far from ideal. For example, a popular morphable model of a human face [66], shown in Figure 1·5, lacks many features of real faces, such as hair, subsurface scattering, or cast shadows. In this thesis, we show that, despite these limitations, we can transfer control over individual factors that can be manipulated in this simulation, such as the facial expression or head orientation, onto a real face domain using unsupervised adversarial alignment. Finally, we note that while we, of course, can not learn to accurately manipulate individual factors of variation in real images using a synthetic dataset if that factor is absent in that synthetic dataset, like hair color in the example above, in this thesis we show that we can still learn to differentiate such "missing" factors from those shared across two domains without any pair supervision and manipulate such domain-specific and shared attributes in isolation.

More specifically, in Chapter 3, we address the stability and semantic consistency of unsupervised adversarial alignment. After that, in Chapter 4, we focus on the task of controlled manipulation of visual attributes of real images using cross-domain supervision in two specific setups. In the "cross-domain image manipulation by demonstration" [119] setup, discussed in Section 4.1, we show how to manipulate a single specific attribute of a real image for which we have "demonstrations" of the desired manipulation in the synthetic domain. For example, the task might be to learn to manipulate face expressions in photos of real humans using "demonstrations" of face expression manipulations on 3D face renders. In the second "unsupervised multimodal translation" [47] setup, discussed in Section 4.2, we show how to manipulate groups of attributes that are specific to images from the input domain while preserving attributes shared across domains without any explicit attribute supervision. For example, let us assume that males and females look sufficiently different and that in addition to that, males have a variable amount of facial hair, and females have a variable amount of makeup. We would like to learn to control the amount of facial hair in male images and the amount of makeup in female images, while preserving attributes that vary across both domains, such as face orientation. We would like the resulting method to learn this using just two unlabeled sets of images as supervision without relying on any attribute annotations or pair supervision. The contributions made in this thesis can be summarized as follows:

- In Section 3.1 of Chapter 3, we show how to stabilize adversarial alignment by dualizing the discriminator objective if the discriminator is a logistic one and show its relation to the maximum mean discrepancy [38] and iteratively-reweighted least squares. This work was presented at the International Conference on Machine Learning (ICLR) workshop [118].

- In Section 3.2 of Chapter 3, we show how to stabilize adversarial alignment with a much richer family of discriminators if the learned transformation is a normalizing flow. We show how to bound the adversarial alignment objective with a novel non-adversarial objective, and show its relation to Jensen-Shannon divergence and GANs [35]. This work was presented at the Advances in Neural Information Processing Systems (NeurIPS) conference [120].

- In Section 3.3 of Chapter 3, we investigate the effect that the cycle-consistency loss (a regularization technique proposed by Zhu et al. [127] and used in the vast majority of state-of-art unsupervised image alignment methods since then) has on the semantic consistency of learned cross-domain mappings. We show that all methods that use this loss end up learning to "cheat" by performing an adversarial attack "on themselves" to satisfy this loss. We show that this cheating manifests as a low-amplitude structured noise in translated images and that the semantic consistency of the learned cross-domain image mapping can be improved by defending against this adversarial attack using a new adversarial objective. This work was presented at the Advances in Neural Information Processing Systems (NeurIPS) conference [7].

- In Section 4.1 of Chapter 4, we show how to train a model that can manipulate a specific attribute of an input image if we have access to triplets of images "demonstrating" this manipulation in a different domain. For example, the proposed model can learn to realistically manipulate mouth expression or lighting in images of real humans using demonstrations of how these manipulations look on 3D renders. This work was presented at the International Conference on Computer Vision (ICCV) [119].

- In Section 4.2.2 of Chapter 4, we demonstrate all existing translation models that claim to be able to infer which attributes are domain-specific and which are shared in both domains using only unpaired examples from two domains, actually mostly rely on biases hard-coded into their architectures, that enable them to work well on some datasets but make them fail miserably on others. This work was presented at the Winter Conference on Applications of Computer Vision (WACV) [8].
- In Section 4.2.3 of Chapter 4, we show how to manipulate attributes specific to one domain while preserving attributes present in both domains (or vice versa) without any explicit attribute supervision. The proposed method infers which attributes are specific for each domain from data using self-adversarial defenses described in Sec. 3.3. This work [121] is available as a technical report on arXiv.

Author Contribution. Ben Usman is the second author on two papers [7, 8] discussed above. His contribution to these papers is limited (see Sec. 3.3 and 4.2.2).

Chapter 2

Background

In this section, we give a brief overview of the unsupervised adversarial alignment and the background necessary for the remainder of this thesis. For a more in-depth discussion of prior work relevant specifically to stabilization and cross-domain disentanglement, please refer to the background subsections of corresponding chapters.

Neural networks. In 1806, Legendre [64] derived a way to estimate functional relationships between covariates $\mathbf{X}$ and outcome variables $\mathbf{Y}$ from observations, assuming that the inferred relationship is linear up to a normally distributed residual ε:

$$\mathbf{Y} = \beta \cdot \mathbf{X} + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma) \tag{2.1}$$

Later generalized linear models [88] relax this assumption, allowing estimation of relationships that are given by a composition of an unknown linear and a fixed known functional relation $g(x)$, up to a residual ε that follows some fixed known distribution $\mathcal{P}$ from the exponential family:

$$\mathbf{Y} = g^{-1}(\beta \cdot \mathbf{X}) + \varepsilon,\ \varepsilon \sim \mathcal{P} \tag{2.2}$$

For example, logistic regression is a generalized linear model with $g(x) = \ln(x/(1-x))$ and logistically distributed residuals. Unlike linear models, parameters β of generalized linear models can not be inferred from data in closed-form, and are estimated by solving a sequence of least squares problems iteratively reweighted by the derivatives of $g(x)$. In the seminal work on backpropagation of error, Rumelhart et al. [102] proposed a general algorithm for estimating the parameters of functional relationships given by arbitrary length sequences of linear transformations interspersed with fixed functional relationships, colloquially known as **deep neural networks**:

$$\hat{\mathbf{Y}}(\mathbf{X}, \beta_1, \ldots, \beta_N) = g_N(\beta_N \cdot \ldots g_2(\beta_2 \cdot g_1(\beta_1 \cdot \mathbf{X})) \ldots) \tag{2.3}$$

$$\mathbf{Y} = \hat{\mathbf{Y}}(\mathbf{X}, \beta_1, \ldots, \beta_N) + \varepsilon,\ \varepsilon \sim \mathcal{P} \tag{2.4}$$

Assuming that the negative log-likelihood of the residual ε is given by the function $L(\varepsilon)$, Rumelhart et al. [102] show that the parameter vectors $\beta_1, \ldots, \beta_N$ can be computationally efficiently estimated via **gradient decent** with learning rate α:

$$l(\beta_1, \ldots, \beta_N) = L(\mathbf{Y} - \hat{\mathbf{Y}}(\mathbf{X}, \beta_1, \ldots, \beta_N)) \tag{2.5}$$

$$\beta_i^{(t+1)} = \beta_i^{(t+1)} - \alpha \cdot \left. \frac{\partial l(\beta_1, \ldots, \beta_N)}{\partial \beta_i} \right|_{(\beta_1^{(t)}, \ldots, \beta_N^{(t)})} \tag{2.6}$$

The resulting algorithm for computationally efficiently updating parameters of deep neural networks is called the **backpropagation** algorithm. Auto-differentiation with reverse-mode accumulation [69] generalizes the backpropagation algorithm to arbitrary directed acyclic graphs of computation, allowing multiple branches of independent computations to share parameters and minimize multiple objectives.

Deep convolutional neural networks. The seminal work of LeCun et al. [60] showed that replacing linear transformation with inherently translation-invariant multichannel convolutions, and adding spatial pooling and sub-sampling in-between

$$\hat{\mathbf{Y}}(\mathbf{X}, \mathbf{w}_1, \mathbf{w}_2, \beta_3, \beta_4, \beta_5) = \beta_5 \cdot \tanh(\beta_4 \cdot \tanh(\beta_3 \cdot \mathrm{flatten}(\mathbf{h}_2))) \tag{2.7}$$

$$\mathbf{h}_2 = \mathrm{pool}(\tanh(\mathbf{w}_1 * \mathbf{h}_1)), \quad \mathbf{h}_1 = \mathrm{pool}(\tanh(\mathbf{w}_1 * \mathbf{X})) \tag{2.8}$$

results in better generalization to unseen data on tasks involving **images**, and pro-

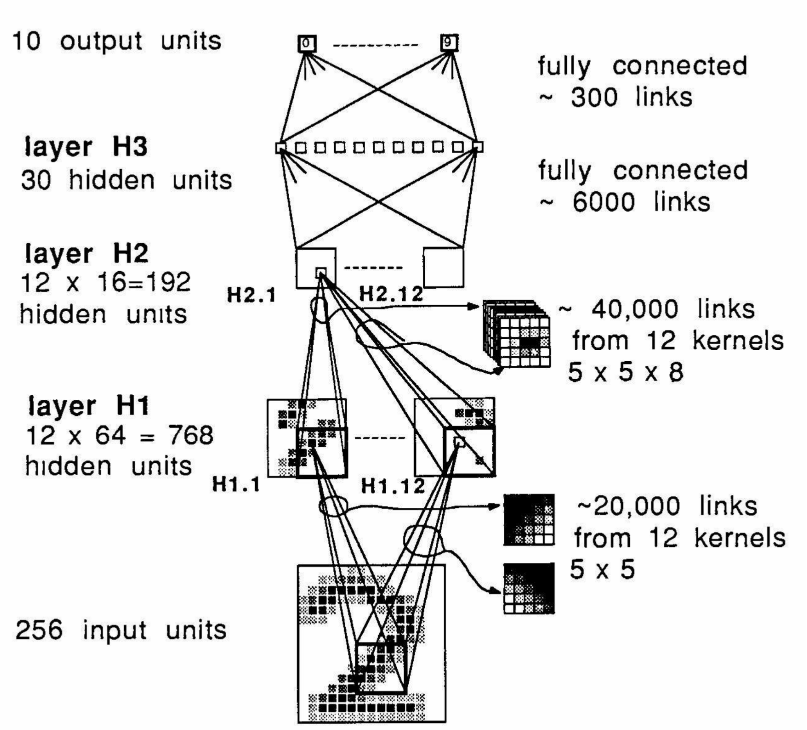


Figure 2·1: LeNet-5 [60] architecture and activation maps.

duces interpretable spatial **feature maps** $\mathbf{h}_1$ and $\mathbf{h}_2$, see Fig. 2·1. Since then, many critical improvements were made to this architecture, including (but not limited to) residual connections [42] that enabled training deeper networks, and batch normalization [48] that improved convergence of large networks, but the overall perspective of seeing deep convolutional networks as sequences of translation-invariant non-linear transformations applied to spatial feature maps remained largely intact.

Deep convolutional generators. In many real-world tasks, such as colorization [16], super-resolution [79], or surface normal estimation [123], the estimated underlying function takes an image as an argument and produces another image as an output. While nothing prevents us from regressing individual pixel intensities independently from each other, similar to how we regressed outcome variables in Eq. (2.7), approaches that take into account the spatial structure of generated images tend to produce higher quality results. Most approaches factorize the learned image-to-image mapping $f : x \to y$ into a deep **convolutional encoder** $E : x \to e$ mapping an input image x into an intermediate embedding e, architecturally similar to deep convolutional networks presented in the previous paragraph, and a **deconvolutional generator** $G : e \to y$ mapping the embedding vector e to the output image y, *i.e.* $f = E \circ G$. The architecture of the generator, introduced under a different name by Zeiler et al. [126], mirrors the architecture of the encoder, but in reverse: replacing pooling with upsampling, and convolutions with transposed convolutions [126].

Unsupervised adversarial alignment. As discussed in the previous chapter, our ultimate goal is to learn a relationship between two visual domains from unpaired examples from these domains. The first step towards this goal is to learn the vector of parameters θ of an encoder-generator transformation $T(x; \theta)$ that would map images from one domain into plausible examples from another. The parameter vector θ is usually updated via gradient descent to minimize some notion of distinguishability between examples from respective domains, *i.e.* to perform unsupervised domain alignment. This distinguishability is often defined as a maximum possible difference between empirical means of values of a *witness function* $f(x)$ evaluated on samples A and B from respective domains, maximized over the function family $\mathcal{F}$:

$$d_f(A, B) = \max_{f \in \mathcal{F}} \left(\mathbb{E}_{x \sim A}\ f(x) - \mathbb{E}_{x \sim B}\ f(x)\right) \tag{2.9}$$

An example of such witness function for a pair of a Gaussian and a Laplacian distribution is given in Figure 2·2. Starting from the seminal work of Goodfellow et al. [35] introducing generative adversarial networks (GANs), this distinguishability is often defined as the inverted classification loss $L(\hat{p}, p)$ of the best deep convolutional classifier $\hat{p} = f(x; w)$ with parameter vector w, called **discriminator**, and trained to discriminate real examples from B from transformed examples from A. This way,

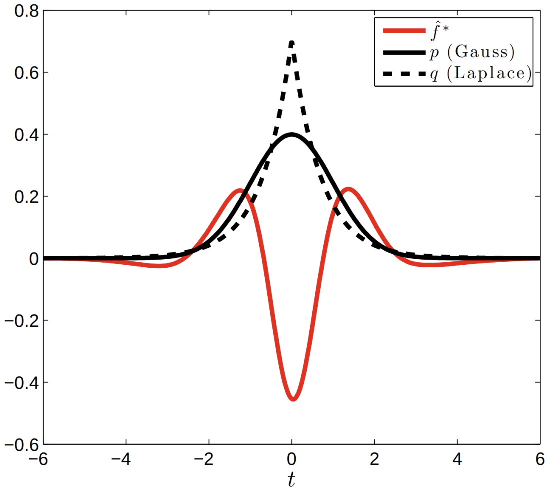


Figure 2·2: The optimal witness function f discriminating samples from the Gaussian p and Laplace distribution q. *(Gretton et al. [38])*

in order to align distributions A and B, one has to minimize the following objective with respect to the parameters θ of the learned transformation:

$$\min_{\theta} d_f(T(A;\theta), B) = \min_{\theta} \max_{w} \left(\mathbb{E}_{x\sim A}\ L(f(T(x;\theta);w);1) - \mathbb{E}_{x\sim B}\ L(f(x;w);0)\right) \tag{2.10}$$

Supervised image-to-image translation. Isola et al. [49] were among the first to show that the adversarial alignment objective (2.10) can improve the visual quality in the supervised image translation task. Figure 2·3 shows that for a **paired** dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=0}^{N}$ of, for example, shoes and corresponding shoe outlines, the task is to learn the parameters of the mapping $T(x;\theta)$ that maps a novel outline into a realistic shoe. For a generated shoe image $\hat{y}_i = T(x_i;\theta)$, Isola et al. [49] optimized the combination of a supervised L1 loss, and an unsupervised alignment loss ensuring that the generated shoe-outline pair looks "indistinguishable" from the real pair:

$$L_{\text{pix2pix}}(x_i, y_i, \hat{y}_i) = \sum_i \|\hat{y}_i - y_i\|_1 + d_f((\hat{y}_i, x_i), (y_i, x_i)). \tag{2.11}$$

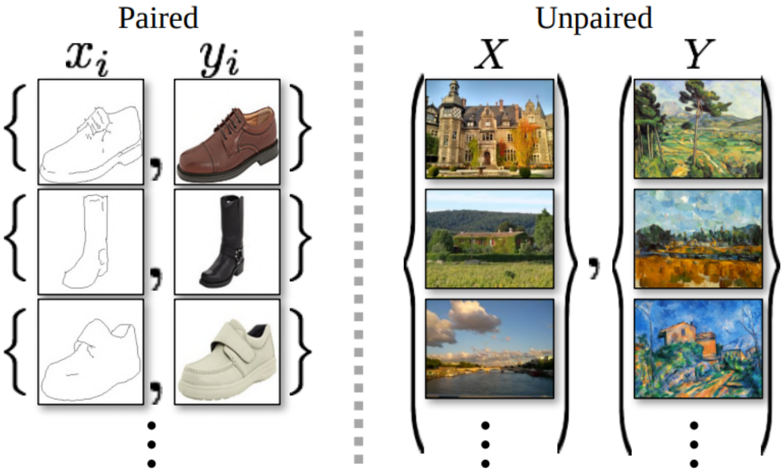


Figure 2·3: The comparison between supervised and unsupervised image translation tasks (from the work of Zhu et al. [127]).

Unsupervised image-to-image translation. CycleGAN [21] and UNIT [71] were the first to show that adversarial domain alignment objective (2.10) can be used to infer relationships between complex distributions in higher dimensions, such as distributions of natural images, even **without** any pair supervision (see Figure 2·3). In order to achieve high-quality alignment, they trained two separate mappings ($T(x; \theta_{\text{A2B}})$ and $T(x; \theta_{\text{B2A}})$) and applied alignment losses in both domains

$$L_{\text{adv}} = d_f(T(A; \theta_{\text{A2B}}), B) + d_f(T(B; \theta_{\text{B2A}}), A) \tag{2.12}$$

and regularized learned mappings either by weight sharing, or via a VAE-like [56] KL-penalty terms [71], or the cycle-consistency loss

$$L_{\text{cyc}} = \mathbb{E}_{a \sim A} \, \|T(T(a; \theta_{\text{A2B}}); \theta_{\text{B2A}}) - a\| + \mathbb{E}_{b \sim B} \, \|T(T(b; \theta_{\text{B2A}}); \theta_{\text{A2B}}) - b\|, \tag{2.13}$$

or a combination of the above. We refer to models that use cycle-consistency loss (2.13) as **cycle-consistent models** in the remainder of this thesis. These methods were shown [21, 71] to be able to capture semantic correspondances in image layouts across visual domains and realistically translate between, for example, images of outdoor scenes captured at daytime and nighttime, images of horses and zebras, and even images of driving scenes and so called “segmentation maps” specifying only types, approaching (but not quite reaching) the performance of supervised image-to-image translation methods.

Domain Adaptation. Neural domain adaptation methods seek to improve the performance of a classifier network on a target distribution that is different from the original training distribution by introducing an additional objective that minimizes the difference between representations learned for source and target data, similar to the unsupervised alignment objective (2.10). Some models align feature representations across domains by minimizing the distance between first or second-order feature space statistics [76, 109, 115]. When adversarial objectives are used for domain adaptation, a domain classifier is trained to distinguish between the generated source and target representations, either using the standard GAN objective [30], or an alternative adversarial objective [116, 117]. Ben-David et al. [10] showed that the test error of the learning algorithm trained and tested on samples from different distributions labeled using a shared “ground truth” labeling function is bounded by the $\mathcal{H}\Delta\mathcal{H}$-distance between the two distributions, therefore framing domain adaptation as distribution alignment. This particular distance is difficult to estimate in practice, so early neural feature-level domain adaptation methods such as deep domain confusion [115], DAN [76], or JAN [77] directly optimized estimates of non-parametric statistical distances (e.g. maximum mean discrepancy) between deep features of data points from two domains. Other early neural DA methods approximated domain distributions via simple parametric models, for example, DeepCORAL [109] minimizes KL-divergence between pairs of Gaussians. Unfortunately, these approaches struggle to capture the internal structure of real-world datasets. Adversarial (GAN-based) approaches, such as ADDA [31] and DANN [117], address these limitations using deep convolutional domain discriminators. However, adversarial models are notoriously hard to train and provide few automated domain-agnostic convergence validation and model selection protocols, unless ground truth labels are available. Many recent improvements in the performance of classifiers adapted using adversarial alignment rely on techniques utilizing source labels, such as semantic consistency loss [44], classifier discrepancy loss [104], or pseudo-labeling [29], added on top of the unsupervised adversarial alignment.

Challenges in adversarial alignment. The instability of adversarial training was pointed out as one of the major factors limiting their wider adoption, often attributed to the fact that training a discriminator until convergence results in vanishing and noisy generator gradients [2]. This issue was tackled from many different directions [84] including closed-form discriminator regularization in [100] and instance noise and data augmentations [54], as well as better discriminator objectives that can be optimized until convergence without causing the vanishing of generator gradients [2, 86].

Normalizing Flows [98] are a class of unsupervised models that can capture the complexity of high-dimensional distributions and do not suffer from training instability. The main assumption behind such models is that the unknown distribution P_X of observed samples $x \sim P_X$ can be modeled as a simple known distribution P_Z transformed by a sequence of simple unknown invertible transformations $T_1, \ldots, T_M$:

$$z_0 \sim P_Z, \quad z_0 \xrightarrow{T_1} z_1 \xrightarrow{T_2} \ldots \xrightarrow{T_M} x, \quad x \sim P_X \tag{2.14}$$

For a given sequence of transformations, the density at a given point x can be estimated using the change of variable formula by inverting each transformation and evaluating determinants of Jacobians at corresponding points:

$$P_X(x|T_1,\dots,T_M) = P_Z(T_1^{-1}(T_2^{-1}(\dots(T_M^{-1}(x)))) \cdot \prod_j \det\left|\frac{\partial T_j^{-1}(z_j)}{\partial z_j}\right| \tag{2.15}$$

Then the distribution P_X can be estimated from samples $\{x_i\}_i$ via maximum likelihood by optimizing the objective above over parameters of all transformations. The main challenge in developing such models is to define a class of atomic transformations T that are invertible, rich enough to model real-world distributions, and simple enough to enable direct estimation of the aforementioned Jacobian determinant. Most notable examples of recently proposed normalizing flows include Real NVP [25], GLOW [57] built upon Real NVP with more general learnable permutations and trained at multiple scales to handle high-resolution images, and the recent FFJORD [36], which used forward simulation of an ODE with a velocity field parameterized by a neural network as a flow transformation. In the next chapter, we show how normalizing flows can be used to stabilize adversarial domain alignment.

In this thesis, we show that the min-max nature of the alignment objective (2.10) causes unstable training and that successful alignment alone is not sufficient for controlled manipulation of real images. In the following chapters, we show several ways of stabilizing adversarial alignment using objective dualization and normalizing flows, improving the semantic consistency of the alignment, and applying these ideas to controlled manipulation of natural images with cross-domain supervision.

Chapter 3

Stability and Semantic Consistency of Adversarial Alignment

As discussed in the previous chapter, mathematically, adversarial domain alignment requires solving a saddle point problem (2.10). As we demonstrate with a simple example in Section 3.1.1, using gradient descent to solve saddle point problems is inherently very difficult. In Section 3.1 we explore how restricting the problem to a logistic discriminator, and dualizing the resulting logistic objective affects the stability of the resulting alignment. In Section 3.2 we show that even for a much more general class of discriminators, the inner maximization problem can be bounded by a minimization problem, if the learned domain transformation is a normalizing flow, effectively reducing the min-max problem to a minimization problem with known convergence guarantees. In Section 3.3 we show that the cycle loss, often used to improve semantic consistency of learned domain correspondences, causes embedding of the low-amplitude structured noise into intermediate generated images, and propose a new adversarial loss that prevents this "cheating" and, as a result, improves the translation accuracy.

3.1 Stabilizing Alignment via Objective Dualization

In this section, we explore how replacing the maximization part of the adversarial alignment problem with a dual minimization problem for a logistic discriminator affects the stability of the alignment. Moreover, we show that it is strongly related

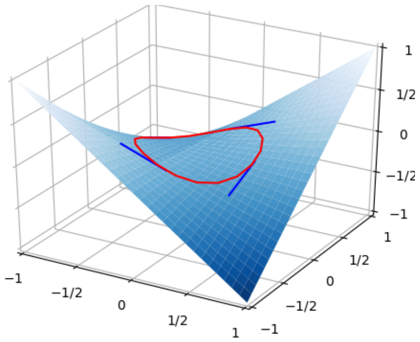


Figure 3·1: Gradient descent fails to solve the saddle point problem $\min_x \max_y xy$. The red line shows the trajectory of the gradient descent if vector field $g(x, y) = (y, -x)$ is used at each iteration. Blue lines are examples of vectors from this vector field.

to the iteratively reweighted empirical estimator of maximum mean discrepancy [38]. We also evaluate how well our dual method can handle a point alignment problem on a low-dimensional synthetic dataset, and compared its performance with the analogous primal method on a real-image domain adaptation problem using the Street View House Numbers (SVHN) and MNIST domain adaptation dataset pair. In these experiments, the goal is to align feature distributions produced by the network on the two datasets so that a classifier trained to label digits on SVHN does not lose accuracy on MNIST due to the domain shift. In both cases, we show that the proposed dual formulation of the adversarial distance yields consistent improvement over time, whereas using the primal formulation results in unstable training and often does not converge.

3.1.1 Motivating Example

We start with a well-known [35] motivating example of a simple min-max problem to show that, even in this simple case, gradient descent fails dramatically. Let us consider the problem of finding a saddle point of a hyperbolic surface. Given the function $f(x, y) = xy$, our problem is to solve $\min_x \max_y f(x, y)$, which has a unique solution at $(0,0)$. Suppose that we want to apply gradient descent to solve this problem. The intuitive analog of the gradient vector that we might consider using in the update rule is defined by the vector field $g(x,y) = (x, -y)$. However, at any given point the vector $g(x,y)$ will be tangent to a closed circular trajectory, thus following this trajectory would never lead to the true solution $(0,0)$. One can observe the trajectory produced by the update rule $x_{t+1} = x_t + \alpha g(x,y)$ applied to the problem above in Figure 3·1. Neither block coordinate descent nor various learning rate schedule can improve the performance of the gradient descent on this problem.

3.1.2 Background

Other objectives for distribution matching that have been proposed in the literature, including Maximum Mean Discrepancy [38], f-discrepancy [92], and others, have also been used for generative modeling [26, 67]. A single step of our iterative reweighting procedure is similar to instance reweighting methods that were theoretically and empirically shown to improve accuracy in the presence of domain shift. For example, Huang et al. [45] used sample reweighting that minimized empirical MMD between populations to plug it as instance-weights in weighted classification loss, whereas Gong et al. [34] did that to choose points for a series of independent auxiliary tasks, so no *iterative* reweighting was performed in both cases. In an independent and concurrent work, Li et al. [68] proposed to dualize the local linear approximation of the min-max objective to stabilize the procedure.

In general, most statistical distances used for distribution alignment fall into one of two categories: they are either f-divergences (e.g. GAN objective, KL-divergence), or integral probability metrics (IPMs) that are differences in expected values of a test function at samples from different distributions maximized over some function family (e.g. Maximum Mean Discrepancy, Wasserstein distance). In this work, we specifically consider the logistic adversarial objective (f-divergence), show that it is useful to optimize its dual, and present a relation between this adversarial dual objective and MMD, another statistical distance with a test function from reproducing kernel Hilbert space (RKHS).

While there is a huge body of work on using alternative descent schemes for convex-concave saddle point optimization, including, but not limited to different variants of mirror descent, such as Nesterov's dual averaging [90] and Mirror Prox [89], authors are not aware of any successful attempts to use it in the context of adversarial distribution alignment, likely because the problem at hand is rarely convex-concave. Some techniques developed for solving continuous games such as fictitious play were successfully adopted by Salimans et al. [105].

3.1.3 Dual Logistic Adversarial Distance

We first propose a new formulation of the adversarial objective for distribution alignment problems. Then we apply this approach to the domain adaptation scenario in Section 3.1.3. Suppose that we are given a finite set of points A sampled from the distribution p, and a finite set of points B sampled from the distribution q, and our goal is to match q with p by aligning B with A. More specifically, we aim to learn a matching function $F_\theta(B)$ that maps B to be as close as possible to A by minimizing some empirical estimate of a statistical distance $d(\cdot,\cdot)$ between them where θ are parameters of the matching function: $\theta^* = \operatorname{argmin}_\theta d(A, F_\theta(B))$.

Let us denote $B'_\theta = F_\theta(B)$ or just B' in contexts where dependence on θ is not important. The regular adversarial approach obtains the distance function by finding the best classifier $D_w(x)$ with parameters w that discriminates points $x \in A$ from points $x \in B'$ and considers the distance between A and B' to be equal to the likelihood of this classifier. A higher likelihood of separating A from B' means that A is far from B'. This can be any form of hypothesis in general and is often chosen to be a linear classifier [115] or a multi-layer neural network [35]. In this work, we use the class of linear classifiers, specifically, logistic regression in its primal and dual formulations. The solution can also be kernelized to obtain nonlinear discriminators.

We will define the distance between distributions to be equal to the maximum likelihood of the logistic classifier parametrized by w:

$$d(A, B') = \max_w \sum_{x_i \in A} \log(\sigma(w^T x_i)) + \sum_{x_j \in B'} \log(1 - \sigma(w^T x_j)) - \frac{\lambda}{2} w^T w \tag{3.1}$$

We can equivalently re-write this expression as:

$$C_\theta = \{(x_i, y_i) \ : \ x_i \in A \cup B'_\theta \ , \ y_i = 1 \ \text{ if } \ x_i \in A \ \text{ else } \ -1\} \tag{3.2}$$

$$\min_\theta d(A, B'_\theta) = \min_\theta \max_{w,b} \sum_{x_i, y_i \in C_\theta} \log(\sigma(y_i(w^T x_i + b))) - \frac{\lambda}{2} w^T w \tag{3.3}$$

The duality derivation [50, 85] follows from the fact that the log-sigmoid has a sharp upper-bound

$$\log(\sigma(u)) \leq \alpha^T u + H(\alpha) \ , \quad \alpha_i \in [0, 1] \tag{3.4}$$

$$H(\alpha) = \alpha^T \log \alpha + (1 - \alpha)^T \log(1 - \alpha) \tag{3.5}$$

thus we can upper-bound the distance as

$$d(A, B'_\theta) = \min_{0 \leq \alpha \leq 1} \max_{w,b} \sum_{x_i, y_i \in C_\theta} \alpha_i y_i (w^T x_i + b) + H(\alpha) - \frac{\lambda}{2} w^T w \tag{3.6}$$

where the dual variable α_i corresponds to the weight of the point x_i. A higher weight means that the point is contributing more to the decision hyperplane. The optimal value of alpha attains this upper bound. The w that maximizes the inner expression can be computed in a closed form as $w^* = \frac{1}{\lambda}(\sum_j x_j y_j \alpha_j)$. Optimally of bias requires

$\sum_i \alpha_i y_i = 0$. By substituting w^* we obtain a minimization problem:

$$d(A, B'_\theta) = \min_{0 \le \alpha_i \le 1} \ \frac{1}{2\lambda} \sum_{ij} \alpha_i \alpha_j (y_i x_i)^T (y_j x_j) + H(\alpha) = \min_{0 \le \alpha_i \le 1} \ \frac{1}{2\lambda} \alpha^T Q \alpha + H(\alpha) = \tag{3.7}$$

$$= \min_{0 \le \alpha_i \le 1/\lambda} \ \frac{1}{2} \alpha_A^T Q_{AA} \alpha_A + \frac{1}{2} \alpha_B^T Q_{BB} \alpha_B - \alpha_A^T Q_{AB} \alpha_B + H(\alpha_A) + H(\alpha_B) \tag{3.8}$$

$$\text{s.t. } ||\alpha_A||_1 = ||\alpha_B||_1$$

The equation (3.8) is obtained by splitting the summation into blocks that include samples only from A, only from B, and from both A and B. For example, the matrix $Q_{AB} = A^T B$ consists of pairwise similarities between points from A and B, and is equal to the dot product between corresponding data points in the linear case and kernel similarity in the kernelized case. The factor of two in front of the cross term comes from the fact that off-diagonal blocks in the quadratic form are equal. The constraint on alpha sums comes from splitting optimality conditions on the bias into two term. We will denote the resulting objective as $d_D(\alpha, A, B)$.

The above expression gives us a tight upper bound on the likelihood of the discriminator. Thus, by minimizing this upper bound, we can minimize the likelihood itself, as in the original loss, and therefore minimize the distance between the distributions:

$$\theta^*, \alpha^* = \operatorname*{argmin}_{\theta, \alpha \in \mathcal{A}} d_D(\alpha, A, F_\theta(B)) \tag{3.9}$$

Note that the overall problem has changed from an unconstrained saddle point problem to a smooth constrained minimization problem, which ultimately converges when gradient descent has a properly chosen learning rate, whereas the descent iterations for the saddle point problem are not guaranteed to converge at all.

The resulting smooth optimization problem consists of minimization over α to improve classification scores and over θ to move points towards the decision boundary.

The next section provides more intuition behind the resulting iterative procedure.

Relationship to MMD. In this section, we show that our dual formulation of the adversarial objective has an interesting relationship to another popular alignment objective. The integral probability metric between distributions p and q with a given function family $\mathcal{H}$ is defined as

$$d(p, q) = \sup_{f \in \mathcal{H}} \Big| \mathbb{E}_{x \sim p} f(x) - \mathbb{E}_{x \sim q} f(x) \Big|. \tag{3.10}$$

It was shown to have a closed form solution and a corresponding closed form empirical estimator if $\mathcal{H}$ is a unit ball in reproducing kernel Hilbert space with the reproducing kernel $k(x, y)$ and is commonly referred to as Maximum Mean Discrepancy [38]:

$$d(p, q) = \frac{1}{2}\mathbb{E}_{p \times p} k(x, x') + \frac{1}{2}\mathbb{E}_{q \times q} k(y, y') - \mathbb{E}_{p \times q} k(x, y) \tag{3.11}$$

$$d(A, B) = \frac{1}{2|A|} \sum_{i,j \in A} k(x, x') + \frac{1}{2|B|} \sum_{i,j \in B} k(y, y') - \frac{1}{|A||B|} \sum_{A \times B} k(x, y). \tag{3.12}$$

From the definition, it is essentially the distance between means of vectors from p and q embedded into the corresponding RKHS. The resulting empirical estimator combines average inner and outer similarities between samples from the two distributions and goes to zero as the number of samples increases if $p = q$.

Note that if sample weights in Eq. (3.8) are constant and equal across all samples, so $\alpha_i = c$, then the dual distance introduced above becomes exactly an empirical estimate of the MMD plus the constant from the entropic regularizer. Thus, the adversarial logistic distance introduced in Eq. (3.8) can be viewed as an *iteratively reweighted* empirical estimator of the MMD distance. Intuitively, what this means is that the optimization procedure consists of two alternating minimization steps: (1) find the best sample weights assignment by changing α so that the regularized weighted MMD is minimized, and then (2) use a fixed α to minimize the resulting *weighted* MMD distance by changing the matching function F_θ. This makes the resulting procedure similar to the Iteratively Reweighted Least Squares Algorithm [37] for logistic regression. An interesting observation here is that it turns out that high weights in this iterative procedure are given to the most mutually close subsets of A and B', where closeness is measured in terms of Maximum Mean Discrepancy. These happen to be exactly the support vectors of the corresponding optimal domain classifier. Therefore, the procedure described above essentially brings sets of the support vectors of the optimal domain classifier from different domains closer together.

We note that the computational complexity of a single gradient step of the proposed method grows quadratically with the size of the dataset because of the kernelization step. However, our batched GPU implementation of the method performed on par with MMD and outperformed primal methods, probably because inference in modern neural networks requires so many dot products that a batch size × batch size multiplication is negligibly cheap compared to the rest of the network with modern highly parallel computing architectures.

Domain Adaptation. We now show how the above formulation can be applied to unsupervised domain adaptation. In this scenario, we train our classifier in a supervised fashion on some domain A and have to update it to perform well on a different domain B without using any labeled samples from the latter. Common examples include adapting to a camera with different image quality or to different weather conditions.

More rigorously, we assume that there exist two distinct distributions on $\mathcal{X} \times \mathcal{Y}$: a source distribution $P_S(X, Y)$ and a target distribution $P_T(X, Y)$. We assume that we observe a finite number of labeled samples from the source distribution $D_S \subset [\mathcal{X} \times \mathcal{Y}]^n \sim P_S(X, Y)$ and a finite number of unlabeled samples from the target distribution $D_T \subset [\mathcal{X}]^m \sim P_T(X)$. Our goal then is to find a labeling function

$f : \mathcal{X} \to \mathcal{Y}$ from a hypothesis space $\mathcal{F}$ that minimizes target risk $\mathcal{R}_T$, even though we only have labels for samples from the source.

$$\mathcal{R}_T(f) = \mathbb{E}_{(x,y)\sim P_T} L(f(x), y) \leq \mathcal{R}_S(f) + d(P_T, P_S) + C(V, n) \tag{3.13}$$

Ben-David et al. [9] showed that, under mild restrictions on probability distributions, the target risk is upper-bounded by the sum of three terms: (1) the source risk, (2) the complexity term involving the dataset size, and the VC-dimensionality of $\mathcal{F}$, and (3) the *discrepancy* between source and target distributions. Thus, in order to make the target risk closer to the source risk, we need to minimize the discrepancy between distributions. They define discrepancy as a supremum of differences in measures across all events in a given σ-algebra: $d(p, q) = \sup_{A\in\Sigma} |p(A) - q(A)|$. Estimation of the indicated expression is hard in practice, therefore it is usually replaced with more computationally feasible statistical distances. The total variation between two distributions and the Kolmogorov-Smirnov test are closely related to the discrepancy definition above and are also often considered to be too strong to be useful, especially in higher dimensions.

Our approach can be applied directly to this scenario if the discrepancy is replaced with an adversarial objective that uses a logistic regression domain classifier. In Section 3.1.4, we consider an instance of this problem where the main task is classification and the hypothesis space corresponds to multi-layer neural networks. We compare the standard min-max formulation of the adversarial objective in Eq. (3.3) with our min-min formulation in Eq. (3.8), and report the accuracy of the resulting classifier on the target domain.

3.1.4 Experiments and Results

Synthetic Distribution Matching. We first test the performance of our proposed approach on a synthetic point cloud matching problem. The data consists of two clouds of points on a two-dimensional plane and the goal is to match points from one cloud with points from the other. There are no restrictions on the transformation of the target point cloud, so F_θ includes all possible transformations and is therefore parameterized by the point coordinates themselves, so the coordinates themselves were updated on each gradient step. We minimized the logistic adversarial distance in primal space by solving the corresponding min-max problem in Eq (3.3) and compared this to maximizing the proposed negative adversarial distance given by the dual of the logistic classifier in Eq (3.8) and the corresponding kernelized logistic classifier with a Gaussian kernel.

As expected, the optimization of distances given by the dual versions of domain classifiers (linear and kernelized) worked considerably better than the same distance given by a linear classifier in the primal form. More specifically, the results in the primal case were very sensitive to the choice of learning rate. In general, the resulting decent iterations for the saddle point problem did not converge to a single solution, whereas both dual versions successfully converged to solutions that matched the two clouds of points both visually and in terms of means and covariances.

We suggest one more intuitive explanation of why the dual procedure might work better, in addition to the fact that optimization problems are just inherently easier than saddle point problems. The decision boundary of the classifier in the dual space is defined implicitly through a weighted average of observed data points, so when these data points move, the decision boundary moves with them. If points move too rapidly and the discriminator explicitly parametrizes the decision boundary, the weights of the discriminator may change drastically to keep up with moving points, leading to the

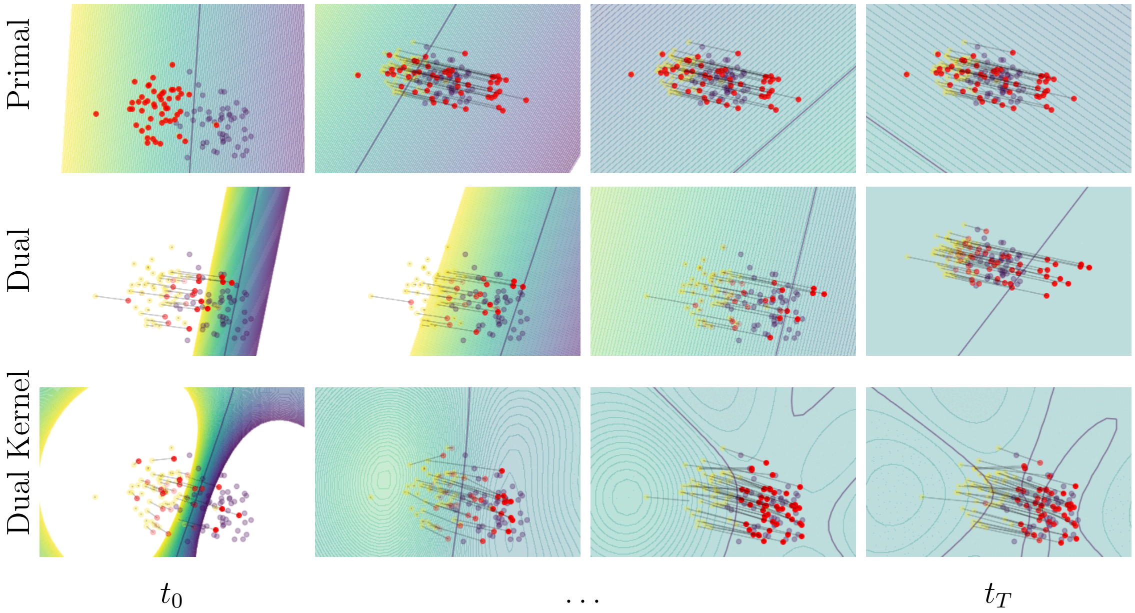


Figure 3·2: (Best viewed in color) When trained on a point cloud matching task, the primal approach leads to an unstable solution that makes the decision boundary *spin* around data points when they are almost aligned, whereas both the linear and kernel dual approaches lead to stable solutions that gradually assign 0.5 probability of belonging to either A or B to all points, which is exactly the desired behavior. Yellow and blue points are the original point clouds, red points correspond to the positions of yellow points after transformation M_θ.

overall instability of the training procedure. In support of this hypothesis, we observed interesting patterns in the behavior of the linear primal discriminator: when point clouds become sufficiently aligned, the decision boundary starts ”spinning” around these clouds, slightly pushing them in corresponding directions. In contrast, both dual classifiers end up gradually converging to solutions that assigned each point with a 0.5 probability of corresponding to either of the two domains.

Feature-Space Unsupervised Domain Adaptation. We also evaluated the performance of the proposed dual objective on a visual domain adaptation task. We performed a series of experiments on an SVHN-MNIST digit classification dataset pair in an unsupervised domain adaptation setup: the task is to use a classifier trained on SVHN and unlabeled samples from MNIST to improve test accuracy on

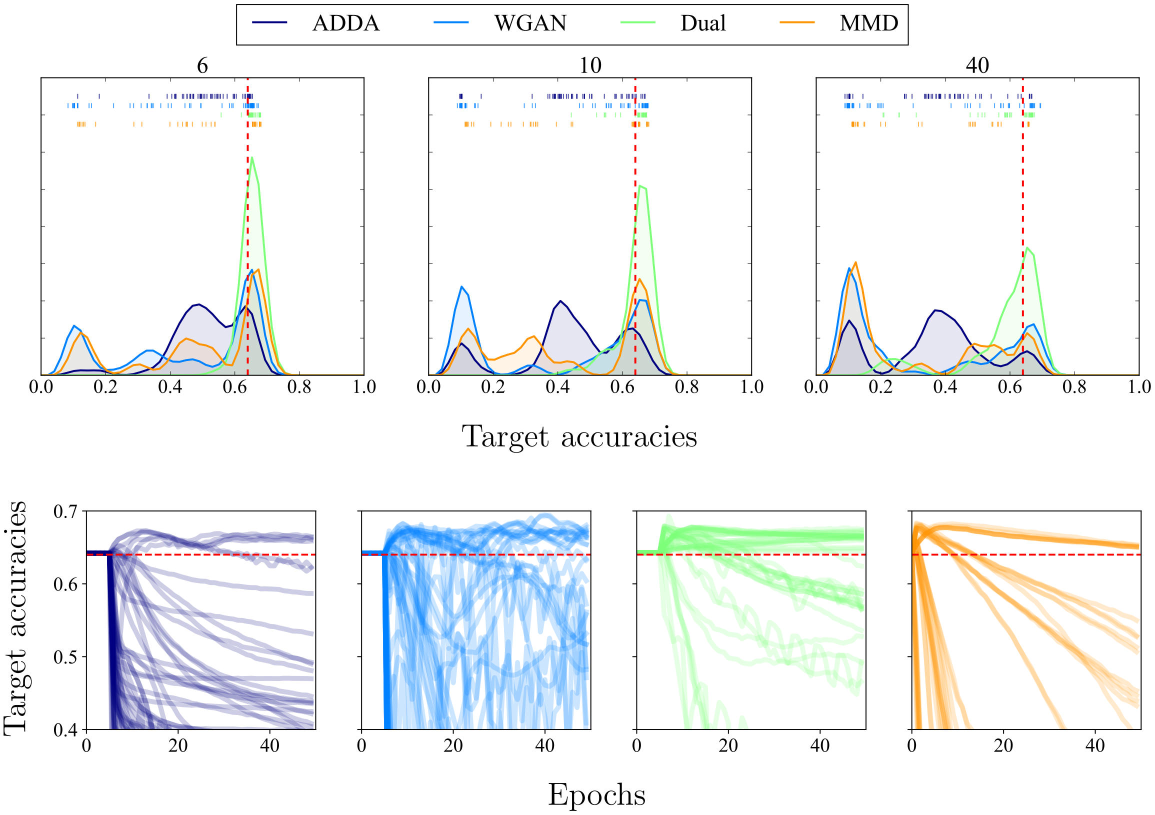


Figure 3·3: (Best viewed in color) **Top row:** Distribution of target test accuracies at different epochs with different objectives during SVHN-MNIST domain adaptation. The red dashed line represents source accuracy, therefore, a larger accuracy distribution mass to the right of (above) the red line is better. These results suggest that our Dual objective leads to very minimal divergence from the optimal solution under the majority of learning rates and hyperparameter combinations. The other methods have lower solution stability, in descending order: Improved WGAN, MMD, ADDA. **Bottom row:** Evolution of target test accuracy over epochs. Our Dual objective (third column) clearly performs well under the majority of the learning rates. WGAN often performs better than MMD and ADDA, but experiences significant oscillations. Different validation heuristics, such as considering only runs that resulted in a significant drop in the distance, did not significantly change these trends. The proportion of runs that outperformed the source baseline after 40 epochs was: 52.3% for Dual, 21.5% for WGAN, 17.1% for MMD, and 6.9% for ADDA.

the latter.

Following Tzeng et al. [117] we used standard LeNet as a base model and outputs of the last layer before the softmax as feature representations. We trained the source network to perform well on source dataset and the discriminator to distinguish features computed by the source and target networks. After training the source network on the source domain, we initialized the target network with source weights and optimized it to make the distributions of source and target feature representations less distinguishable from the discriminator perspective.

We tested several primal objectives based on Adversarial Discriminative Domain Adaptation (ADDA) [117], Improved Wasserstein GAN-based objective with a unit-norm gradient regularizer [105], and MMD [76], and compared them to our Dual objective. To eliminate the influence of a particular discriminator and examine the stability of the *objective structure*, we restricted the discriminator hypothesis space $\mathcal{H}$ to linear classifiers, because primal objectives (ADDA, Improved WGAN) cannot be kernelized and MMD does not support multilayer discriminators. This restriction limits the power of the resulting discriminator, thus leading to scores lower than reported state of the art (usually with carefully chosen hyperparameters), but we are more interested in trends in the behavior of these objectives rather than in absolute reached values.

For each model, we varied learning rates and regularization parameters and ran each experiment for 50 epochs to examine the behavior of these models in the long run. In unsupervised domain adaptation, we do not have access to target labels and thus cannot perform validation of stopping criteria. In fact, if labeled target data were available then it could be used for fine-tuning the source model, rather than just doing unsupervised learning. Therefore we evaluate the behavior of the models over multiple training epochs to see which would be more stable in the face of uncertain stopping criteria in practical domain adaptation scenarios.

Figure 3·3 shows the digit classification accuracies obtained by the four models on the target MNIST dataset. The top row presents the distribution of accuracies at different epochs and the bottom row shows the evolution of individual runs. From these results, we see that on average descent iterations with our Dual objective converged to satisfactory solutions under a considerably higher number of learning rates and hyperparameter combinations compared to other methods. Our model often stayed at peak performance, whereas all other methods most often slowly deviated from it. The amount of instability demonstrates how important it is to choose exactly the right hyperparameters and stopping criteria for these models. In contrast, our Dual objective (third column) clearly performs well under the majority of the learning rates. WGAN often performs better than MMD and ADDA, but experiences significant oscillations. We tried using different validation heuristics, such as considering only runs that resulted in a significant drop in the distance, but this did not significantly change these trends.

Conclusion. In this section we showed that objective dualization leads to more stable optimization without the need for choosing an optimal stopping criterion and learning rates by cross-validation on test data. Unfortunately, this approach is limited only to logistic discriminators. In the next section, we propose an alternative approach that works with a much broader family of discriminators.

3.2 Bounding Likelihood Ratios with Normalizing Flows

Unfortunately, the approach described in the previous section can be applied only if the discriminator family is constrained to logistic classifiers, severely restricting the notion of distinguishability of aligned datasets, and, consequently, the quality of learned alignment. In this section, we discuss how the adversarial objective can be stabilized with a much richer family of discriminators - if the estimated transformation between two domains is a normalizing flow.

The majority of modern neural approaches to domain alignment directly search for a transformation of the dataset that minimizes an empirical estimate of some statistical distance - a non-negative quantity that takes lower values as datasets become more similar. The variability of what “similar” means in this context, which transformations are allowed, and whether data points themselves or their feature representations are aligned, leads to a variety of domain alignment methods. Unfortunately, existing estimators of statistical distances either restrict the notion of similarity to enable closed-form estimation [109], or rely on adversarial (min-max) training [117] that makes it very difficult to quantitatively reason about the performance of such methods [6, 13, 113]. In particular, the value of the optimized adversarial objective conveys very little about the quality of the alignment, which makes it difficult to perform automatic model selection on a new dataset pair. On the other hand, Normalizing Flows [98] are an emerging class of deep neural density models that do not rely on adversarial training. They model a given dataset as a random variable with a simple known distribution transformed by an unknown invertible transformation parameterized using a deep neural network. Recent work on normalizing flows for maximum likelihood density estimation made great strides in defining new rich parameterizations for these invertible transforms [25, 36, 57], but little work focused on flow-based density alignment [39, 125].

In this section, we present the Log-likelihood Ratio Minimizing Flow (LRMF), a new non-adversarial approach for aligning distributions in a way that makes them indistinguishable for a given family of density models M. We consider datasets A and B indistinguishable with respect to the family M if there is a single density model in M that is optimal for both A and B individually since in this case there is no way of telling which of two datasets was used for training it. For example, two different distributions with the same means and covariances are indistinguishable for the Gaussian family M since we can not tell which of two datasets was used by examining the model fitted to either one of them. For a general M, we can quantitatively measure whether two datasets are indistinguishable by models from M by comparing the average log-likelihoods of two "private" density models each fit independently to A and B, to the average log-likelihood of the "shared" model fit to both datasets at the same time. We observe that, if datasets are sufficiently large, the maximum likelihood of the "shared" model would reach the likelihoods of two "private" models on respective datasets only if the shared model is optimal for both of them individually, and consequently datasets are equivalent with respect to M. Then a density model optimal for A is guaranteed to be optimal for B and vice versa.

We want to find a transformation $T(x)$ that transforms dataset A in a way that makes the transformed dataset $T(A)$ equivalent to B for the given family M. We do that by minimizing the aforementioned gap between average log-likelihood scores of "shared" and "private" models.

In this section, we show that, while, in general, such $T(x)$ can be found only by solving a min-max optimization problem, if $T(x, \phi)$ is a family of normalizing flows, then the flow $T(x, \phi^*)$ that makes $T(A, \phi^*)$ and B equivalent w.r.t. M can be found by minimizing a single objective that attains zero upon convergence. This enables automatic model validation and hyperparameter tuning on the held-out set.

To sum up, the novel non-adversarial data alignment method presented in this section combines the clear convergence criteria found in non-parametric and simple parametric approaches and the power of deep neural discriminators used in adversarial models. Our method finds a transformation of one dataset that makes it “equivalent” to another dataset with respect to the specified family of density models. We show that if that transformation is restricted to a normalizing flow, the resulting problem can be solved by minimizing a single simple objective that attains zero only if two domains are correctly aligned. We experimentally verify this claim and show that the proposed method preserves the local structure of the transformed distribution and that it is robust to model misspecification by both over- and under-parameterization. We show that minimizing the proposed objective is equivalent to training a particular adversarial network, but in contrast with adversarial methods, the performance of our model can be inferred from the objective value alone. We also characterize the vanishing of generator gradient mode that our model shares with its adversarial counterparts, and principal ways of detecting it.

3.2.1 Background

Composition of inverted flows. AlignFlow [39] is built of two flow models G and F trained on datasets A and B in the “back-to-back” composition $F \circ G^{-1}$ to map points from A to B. We argue that the structure of the dataset manifold is destroyed if two flows are trained independently, since two independently learned “foldings” of lower-dimensional surfaces into the interior of a Gaussian ball are almost surely “incompatible” and render correspondences between $F^{-1}(B)$ and $G^{-1}(A)$ meaningless. Grover et al. [39] suggests sharing some weights between F and G, but we propose that this solution does not address the core of the issue. Yang et al. [125] showed that PointFlow - a variational FFJORD trained on point clouds of mesh surfaces - can be used to align these point clouds in the $F \circ G^{-1}$ fashion. But the point correspondences found by the PointFlow are again due to the spatial co-occurrence of respective parts of meshes (the left bottom leg is always at the bottom left) and do not respect the structure of respective surface manifolds. Our approach requires 2-3 times more parameters than our composition-based baselines, but in the next section, we show that it preserves the local structure of aligned domains better, and the higher number of trainable parameters does not cause overfitting.

CycleGAN with normalizing flows. RevGAN [122] used GLOW [57] to enforce the cycle consistency of the CycleGAN, and left the loss and the adversarial training procedure unchanged. We believe that the normalizing flow model for dataset alignment should be trained via maximum likelihood since the ability to fit rich models with plain minimization and validate their performance on held-out sets are the primary selling points of normalizing flows that should not be dismissed.

Likelihood ratio testing for out-of-distribution detection. Nalisnick et al. [87] recently observed that the average likelihood is not sufficient for determining whether the given dataset came from the same distribution as the dataset used for training the density model. A recent paper by Ren et al. [97] suggested using the log-likelihood ratio test on LSTMs to *detect* distribution discrepancy in genomic sequences, whereas we propose a *non-adversarial* procedure for *minimizing* this measure of discrepancy using unique properties of normalizing flows.

3.2.2 Log-Likelihood Ratio Minimizing Flow

In this section, we formally define the proposed method for aligning distributions. We assume that $M(\theta)$ is a family of densities parameterized by a parameter vector θ, and to fit a model to a dataset X we maximize the likelihood of X w.r.t. θ.

Intuitively, if we fit two models θ_A and θ_B to datasets A and B independently,

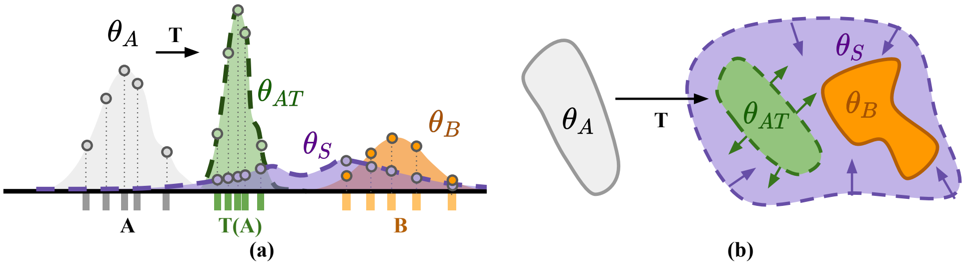


Figure 3·4: To align input datasets A and B, we look for a transformation T that makes $T(A)$ and B "indistinguishable". **(a)** We propose the log-likelihood ratio distance $d_\Lambda(T(A), B)$ that compares likelihoods of density models θ_{AT} fitted to $T(A)$ and θ_B to B independently with the likelihood of θ_S optimal for the combined dataset $T(A) \cup B$. This problem is adversarial, but we show how to reduce it to minimization if T is a normalizing flow. **(b)** Colored contours represent level sets of models for B (orange), $T(A)$ (green), and θ_S (purple), contour sizes corresponds to entropies of these models. Only θ_{AT} and θ_S (dashed) change during training. The proposed objective can be viewed as maximizing the entropy of the transformed dataset while minimizing the combined entropy of $T(A) \cup B$, i.e. expanding the green contour while squeezing the purple contour around the green and orange contours. At equilibrium, θ_S and θ_{AT} model the same distribution as θ_B, i.e. shapes of purple and green contours match and tightly envelope the orange contour. *Best viewed in color.*

and also fit a single shared model θ_S to the combined dataset $A \cup B$, then the log-likelihood ratio distance would equal the difference between the log-likelihood of that optimal "shared" and the two optimal "private" models (Definition 3.2.1).

Next, we consider the problem of finding a transformation that would minimize this distance. In general, this would require solving an adversarial optimization problem (3.16), but we show that if the transformation is restricted to the family of normalizing flows, then the optimal one can be found by minimizing a simple non-adversarial objective (Theorem 3.2.3). We also illustrate this result with an example that can be solved analytically: we show that minimizing the proposed distance between two random variables with respect to the normal density family is equivalent to directly matching their first two moments (Example 3.2.1). Finally, we show the relation between the proposed objective and Jensen-Shannon divergence and show that minimizing the proposed objective is equivalent to training a generative adversarial network with a particular choice of the discriminator family.

Notation. Let $\log P_M(X;\theta) := \mathbb{E}_{x\sim P_X} \log P_M(x;\theta)$ denote the negative cross-entropy between the distribution P_X of the dataset X defined over $\mathcal{X} \subset \mathbb{R}^n$, and a member P_M $(x;\,\theta)$ of the parametric family of distributions $M(\theta)$ defined over the same domain, i.e., the likelihood of X given $P_M(x;\theta)$.

Definition 3.2.1 (Log-likelihood Ratio Distance). *Let us define the log-likelihood ratio distance d_Λ between datasets A and B from $\mathcal{X}$ with respect to the family of densities M, as the difference between log-likelihoods of A and B given optimal models with "private" parameters θ_A and θ_B, and "shared" parameters θ_S:*

$$\begin{aligned} d_\Lambda(A,B;M) &= \max_{\theta_A;\theta_B} \Big[\log P_M(A;\theta_A) + \log P_M(B;\theta_B)\Big] - \max_{\theta_S} \Big[\log P_M(A;\theta_S) + \log P_M(B;\theta_S)\Big] \\ &= \min_{\theta_S} \max_{\theta_A;\theta_B} \Big[\big(\log P_M(A;\theta_A) - \log P_M(A;\theta_S)\big) + \big(\log P_M(B;\theta_B) - \log P_M(B;\theta_S)\big)\Big]. \end{aligned} \tag{3.14}$$

The expression above is also the log-likelihood ratio test statistic $\log \Lambda_n$ for the null hypothesis $H_0 : \theta_A = \theta_B$ for the model described by the likelihood function $P(A, B \mid \theta_A, \theta_B) = \big[P_M(A;\theta_A) \cdot P_M(B;\theta_B)\big]$ and intuitively equals to the amount of likelihood we "lose" by forcing $\theta_A = \theta_B$ onto the model fitted to approximate A and B independently. Figure 3·4 illustrates that, in terms of average likelihood, the shared model (purple) is always inferior to two private models from the same class, unless two datasets are in fact just different samples from the same distribution.

Lemma 3.2.1. *The log-likelihood ratio distance is non-negative, and the equals zero only if there exists a single "shared" model that approximates datasets as well as their "private" optimal models:*

$$d_\Lambda(A,B;M) = 0 \Leftrightarrow \exists\; \theta_S : \log P_M(A;\theta_S) = \max_\theta \log P_M(A;\theta) \wedge \log P(B;\theta_S) = \max_\theta \log P_M(B;\theta). \tag{3.15}$$

Proof. Follows from the fact that the shared part in the Definition 3.2.1 is identical to the private part but over a smaller feasibility set $\{\theta_A = \theta_B\}$. If we define $f(x) =$

$\log P_M(A, x)$ and $g(x) = \log P_M(B, x)$, the first statement $d_\Lambda \geq 0$ follows from the fact that

$$\forall x \; f(x) + g(x) \geq \min_x f(x) + \min_x g(x) \;\; \Rightarrow \min_x (f(x) + g(x)) - \min_x f(x) - \min_x g(x) \geq 0$$

The second statement $f(x^*) = \min_x f(x), g(x^*) = \min_x g(x)$ comes form the fact that the equality holds only if there exists such x^* that

$$f(x^*) + g(x^*) = \min_x f(x) + \min_x g(x)$$

Assume that $f(x^*) \neq \min_x f(x)$, then $f(x^*) > \min_x f(x)$ from the definition of the min, therefore

$$g(x^*) = (f(x^*) + g(x^*)) - f(x^*) < (\min_x f(x) + \min_x g(x)) - \min_x f(x) = \min_x g(x),$$

which contradicts the definition of the $\min_x g(x)$, therefore $f(x^*) = \min_x f(x)$. □

Adversarial formulation. If we introduce the parametric family of transformations $T(x, \phi)$ and try to find ϕ that minimizes the log-likelihood ratio distance $\min_\phi d_\Lambda(T(A; \phi), B; M)$, an adversarial problem arises. Note that for a fixed dataset B, only the first term is adversarial, and only w.r.t. θ_{AT}:

$$\min_{\phi, \theta_S} \max_{\theta_{AT}; \theta_B} \Big[\log P_M(T(A; \phi); \theta_{AT}) + \log P_M(B; \theta_B) - \log P_M(T(A; \phi); \theta_S) - \log P_M(B; \theta_S) \Big] \tag{3.16}$$

Figure 3·4b illustrates that minimizing this objective (3.16) over θ_S while maximizing it over θ_{AT} corresponds to minimizing entropy ("squeezing") of the combination of $T(A)$ and B while maximizing entropy of ("expanding") transformed dataset $T(A)$ as much as possible.

Non-adversarial formulation. The adversarial objective (3.16) requires finding a new optimal model θ_{AT} for each new value of ϕ to find the maximal likelihood of the transformed dataset $T(A)$, but Figure 3·4a illustrates that the likelihood of the transformed dataset can be often estimated from the parameters of the transformation T alone. For example, if T uniformly squeezes the dataset by a factor of two, the average maximum likelihood of the transformed dataset $\max_\theta \log P_M(T(A); \theta)$ doubles compared to the likelihood of the original A. In general, the likelihood of the transformed dataset is inversely proportional to the Jacobian of the determinant of the applied transformation. The lemma presented below formalizes this relation taking into account the limited capacity of M, and leads us to our main contribution: the optimal transformation can be found by simply minimizing a modified version of the objective (3.16) using an iterative method of one's choice.

Lemma 3.2.2. *If $T(x; \phi)$ is a normalizing flow, then the first term in the objective (3.16) can be bounded in closed form as a function of ϕ up to an approximation error $\mathcal{E}_{bias}$. The equality in (3.2.2) holds when the approximation term vanishes, i.e. if M approximates both A and $T(A; \phi)$ equally well; P_A is the true distribution of A and $T[P_A, \phi]$ is the push-forward distribution of the transformed dataset.*

$$\max_{\theta_{AT}} \log P_M(T(A; \phi); \theta_{AT}) \leq \max_{\theta_A} \log P_M(A; \theta_A) - \log\det|\nabla_x T(A; \phi)| + \mathcal{E}_{bias}(A, T, M)$$

$$\mathcal{E}_{bias}(A, T, M) \triangleq \max_{\phi} \left[\min_{\theta} \mathcal{D}_{KL}(P_A; M(\theta)) - \min_{\theta} \mathcal{D}_{KL}(T[P_A, \phi]; M(\theta))\right] \tag{3.17}$$

Proof. Overall, we expand likelihoods of combined and shared datasets given best models from M into respective "true" negative entropies and the approximation errors due to the choice of M (KL-divergence between true distributions and their KL-projections onto M). Then we replace the entropy of the transformed dataset with the entropy of the original and the log-determinant of the Jacobian of the applied transformation, noting that $\log\det|\nabla_x T^{-1}(T(A, \phi), \phi)| = \log\det|\nabla_x T(A, \phi)|$.

More specifically, first, we add and remove the true (unknown) entropy of the distribution $H[P_A] = -\mathbb{E}_{a\sim P_A} \log P_A(a)$:

$$\max_{\theta_A} \mathbb{E}_{a\sim P_A} \log P_M(a; \theta_A) = \max_{\theta_A} \left[\mathbb{E}_{a\sim P_A} \log P_A(a) - \mathbb{E}_{a\sim P_A} \log \frac{P_A(a)}{P_M(a; \theta_A)}\right] \tag{3.18}$$

$$= H[P_A] - \min_{\theta_A} \mathbb{E}_{a\sim P_A} \left[\log \frac{P_A(a)}{P_M(a; \theta_A)}\right] = H[P_A] - \min_{\theta} \mathcal{D}_{KL}(P_A; M(\theta)). \tag{$\star$}$$

And then add and remove the (unknown) entropy of the transformed distribution $H[T[P_A, \phi]]$. We also use the change of variable formula $T[P_A](x) = P_A(T^{-1}(x)) \cdot \det|\nabla_x T^{-1}(x)|$, and substitute the expression for $H[P_A]$ from the previous line ($\star$):

$$\begin{aligned}
&\max_{\theta_{AT}} \log P_M(T(A;\phi);\theta_{AT}) = \max_{\theta_{AT}} \mathbb{E}_{a'\sim T[P_A,\phi]} \log P_M(a';\theta_{AT}) \\
&= \max_{\theta_{AT}} \left[\mathbb{E}_{a'\sim T[P_A,\phi]} \log T[P_A](a') - \mathbb{E}_{a'\sim T[P_A,\phi]} \log \frac{T[P_A,\phi](a')}{P_M(a';\theta_{AT})}\right] \\
&= \max_{\theta_{AT}} \Big[\mathbb{E}_{a\sim P_A} P_A(T^{-1}(T(a,\phi),\phi)) + \\
&\qquad\qquad + \log\det|\nabla_x T^{-1}(T(a,\phi),\phi)| - \mathcal{D}_{KL}(T[P_A,\phi]; M(\theta_{AT}))\Big] \\
&= H[P_A] - \log\det|\nabla_x T(A,\phi)| - \min_{\theta} \mathcal{D}_{KL}(T[P_A,\phi]; M(\theta)) \\
&\leq \max_{\theta_A} \log P_M(A;\theta_A) - \log\det|\nabla_x T(A,\phi)| + \mathcal{E}_{bias}(A,T,M).
\end{aligned} \tag{3.19}$$

□

By applying this lemma to the objective (3.16) and grouping together terms that do not depend on θ_S and ϕ, we finally obtain the final objective.

Definition 3.2.2 (Log-likelihood Ratio Minimizing Flow). *Let us define the log-likelihood ratio minimizing flow (LRMF) for a pair of datasets A and B on $\mathcal{X}$, the family of densities $M(\theta)$ on $\mathcal{X}$, and the parametric family of normalizing flows $T(x;\phi)$ from $\mathcal{X}$ onto itself, as the flow $T(x;\phi^*)$ that minimizes $\mathcal{L}_{LRMF}$ (3.2.2), where the constant $c(A,B)$ does not depend on θ_S and ϕ, and can be precomputed in advance.*

$$\begin{aligned}
\mathcal{L}_{LRMF}(A,B,\phi,\theta_S) &= -\log\det|\nabla_x T(A;\phi)| - \log P_M(T(A;\phi);\theta_S) - \log P_M(B;\theta_S) + c(A,B), \\
c(A,B) &= \max_{\theta_A} \log P_M(A;\theta_A) + \max_{\theta_B} \log P_M(B;\theta_B)
\end{aligned} \tag{3.20}$$

Theorem 3.2.3. *If $T(x,\phi)$ is a normalizing flow, then the adversarial log-likelihood ratio distance (3.16) between the transformed source and target datasets can be bounded via the non-adversarial LRMF objective (3.2.2), and therefore the parameters of the normalizing flow ϕ that make $T(A,\phi)$ and B equivalent with respect to M can be found by minimizing the LRMF objective (3.2.2) using gradient descent iterations with known convergence guarantees.*

$$0 \leq d_\Lambda(T(A,\phi), B; M) \leq \min_\theta L_{LRMF}(A,B,\phi,\theta) + \mathcal{E}_{bias}. \tag{3.21}$$

This theorem follows from the definition of d_Λ and two lemmas provided above that show that the optimization over θ_{AT} can be (up to the error term) replaced by a closed-form expression for the likelihood of the transformed dataset if the transformation is a normalizing flow. Intuitively, the LRMF loss (3.2.2) encourages the transformation T to draw all points from A towards the mode of the shared model $P(x, \theta_S)$ via the second term, while simultaneously encouraging T to expand as much as possible via the first term as illustrated in Figure 3·4b. The delicate balance is attained only when two distributions are aligned, as shown in Lemma 3.2.1. The inequality (3.21) is tight (equality holds) only when the bias term is zero, and the shared model is optimal.

The example below shows that the affine log-likelihood ratio minimizing flow between two univariate random variables with respect to the normal density family M corresponds to shifting and scaling one variable to match two first moments of the other, which agrees with our intuitive understanding of what it means to make two distributions "indistinguishable" for the Gaussian family.

Example 3.2.1. *Let us consider two univariate normal random variables A, B with moments $\mu_A, \mu_B, \sigma_A^2, \sigma_B^2$, restrict M to normal densities, and the transform $T(x;\phi)$ to the affine family: $T(x; a, b) = ax + b$, i.e. $\theta = (\mu, \sigma)$ and $\phi = (a, b)$. Using the expression for the maximum log-likelihood (negative entropy) of the normal distribution, and the expression for variance of the equal mixture, we can solve the optimization over $\theta_S = (\mu_S, \sigma_S)$ analytically:*

$$\min_{\mu,\sigma} \mathbb{E}_X \log P(X; \mu, \sigma) = -\frac{1}{2}\log(2\pi e \sigma_X^2) = -\log \sigma_X + C \tag{3.22}$$

$$\min_{\theta_S} \Big[-\log P_M(T(A;\phi); \theta_S) - \log P_M(B; \theta_S) \Big] = \tag{3.23}$$

$$= \log\left(\frac{1}{2}(a^2\sigma_A^2 + \sigma_B^2) - \frac{1}{4}(\mu_A + b - \mu_B)^2\right) - 2C. \tag{3.24}$$

Combining expressions above gives us the final objective that can be solved analytically *by setting the derivatives with respect to a and b to zero:*

$$\log\det|\nabla_x T(A;\phi)| = \log a \qquad \textit{and} \qquad c(A,B) = -\log\sigma_A - \log\sigma_B + 2C, \tag{3.25}$$

$$\mathcal{L}_{LRMF} = -\log a + \log\left(\frac{1}{2}(a^2\sigma_A^2 + \sigma_B^2) - \frac{1}{4}(\mu_A + b - \mu_B)^2\right) - \log\sigma_A - \log\sigma_B \tag{3.26}$$

$$(a^*, b^*) = \arg\min \mathcal{L}_{LRMF}(a,b) \;\Rightarrow\; a^* = \frac{\sigma_B}{\sigma_A}, \quad b^* = \mu_B - \mu_A. \tag{3.27}$$

The error term $\mathcal{E}_{bias}$ equals zero because any affine transformation of a Gaussian is still a Gaussian.

Relation to Jensen-Shannon divergence and GANs. From the same expansion as in the proof of Lemma 3.2.2 and the information-theoretic definition of the Jensen-Shannon divergence (JSD) as the difference between entropies of individual distributions and their equal mixture, it follows that the likelihood-ratio distance (and consequently LRMF) can be viewed as biased estimates of JSD.

$$d_\Lambda(A,B) = 2\cdot\mathrm{JSD}(A,B) - \mathcal{D}_{KL}(A,M) - \mathcal{D}_{KL}(B,M) + 2\cdot\mathcal{D}_{KL}((A+B)/2, M) \tag{3.28}$$

Also, if the density family M is “convex”, in a sense that for any two densities from M their equal mixture also lies in M,

then by rearranging the terms in the definition of the likelihood-ratio distance, and noticing that the optimal shared model is the equal mixture of two densities, it becomes evident that the LRMF objective is equivalent to the GAN objective with the appropriate choice of the discriminator family:

$$\min_T d_\Lambda(T(A), B, M) = \min_T \max_{\theta_{AT},\theta_B} \min_{\theta_S} \left[\log\frac{P_M(T(A);\theta_{AT})}{P_M(T(A);\theta_S)} + \log\frac{P_M(B;\theta_B)}{P_M(B;\theta_S)}\right] \tag{3.29}$$

$$= \min_T \max_{\theta_{AT},\theta_B} \left[\log\frac{P_M(T(A);\theta_{AT})}{P_M(T(A);\theta_{AT}) + P_M(T(A);\theta_B)} + \log\frac{P_M(B;\theta_B)}{P_M(B;\theta_{AT}) + P_M(B;\theta_B)} + \log 4\right]$$

$$= \min_T \max_{D\in\mathcal{H}} \left[\log D(T(A)) + \log(1 - D(B)) + \log 4\right], \quad \mathcal{H}(\theta,\theta') = \left\{\frac{P_M(x;\theta)}{P_M(x;\theta) + P_M(x;\theta')}\right\}.$$

Since M is not “convex” in most cases, minimizing the LRMF objective is equivalent to adversarially aligning two datasets against a regularized discriminator. From the adversarial network perspective, the reason why $\mathcal{L}_{\text{LRMF}}$ manages to solve this min-max problem using plain minimization is because for any flow transformation parameter ϕ the optimal discriminator between $T(A;\phi)$ and B is defined in closed form: $D^*(x,\phi) = P_M(x;\theta^*_B)/\big(P_M(x;\theta^*_B) + P_M(T^{-1}(x;\phi);\theta^*_A)\det|\nabla_x T^{-1}(x;\phi)|\big)$.

Vanishing of generator gradients. The relation presented above suggests that the analysis performed by Arjovsky and Bottou [2] for GANs (Theorem 2.4, page 6) applies to LRMF as well, meaning that gradients of the LRMF objective w.r.t. the learned transformation parameters might vanish in higher dimensions. This implies that while the inequality (3.21) always holds, the model produces a useful alignment only when a sufficiently "deep" minimum of the LRMF loss (3.2.2) is found, otherwise the method fails, and the loss value should be indicative of this. An example presented below shows that reaching this deep minimum becomes exponentially more difficult as the initial distance between distributions grows, which is often the case in higher dimensions.

Example 3.2.2. *Consider $M(\theta)$ that parameterizes all equal mixtures of two univariate Gaussians with equal variances, i.e. $\theta = (\mu_s^{(1)}, \mu_s^{(2)}, \sigma_s^2)$ and*

$$P_M(x \mid \theta) = \frac{1}{2}\left(\mathcal{N}(x|\mu_s^{(1)},\sigma_s^2) + \mathcal{N}(x|\mu_s^{(2)},\sigma_s^2)\right). \tag{3.30}$$

Consider A sampled from $M(\delta,-\delta,\sigma_0^2)$ and B_μ sampled from $M(\mu+\delta,\mu-\delta,\sigma_0^2)$ for some fixed δ,μ and σ_0. Let transformations be restricted to shifts $T(x;b) = x+b$, so $\phi = b$, and $\log\det|\nabla_x T(x;\phi)| = 0$, and $\mathcal{E}_{bias} = 0$ since M can approximate both A and $T(A;b)$ perfectly for any b. For a sufficiently large μ, optimal shared model parameters can be found in closed form: $\theta^* = (b,\mu,\sigma_0^2+\delta^2)$. This way the LRMF loss can be computed in closed form up to the cross-entropy: $L(b,\mu) := \min_\theta \mathcal{L}_{LRMF}(A,B_\mu,b,\theta) = -2H[M(\mu+\delta,\mu-\delta,\sigma_0^2), M(b,\mu,\sigma_0^2+\delta^2)] + C$. A simulation provided in the Section 3.2.3 shows that the norm of the gradient of the LRMF objective decays exponentially as a function of μ: $\|[\partial L(b,\mu)/\partial\mu](0,\mu)\| \propto \exp(-\mu^2)$, meaning that as A and B_μ become further, the objective quickly becomes flat w.r.t ϕ near the initial $\phi_{t=0} = 0$.

Model complexity. We propose the following intuition: 1) chose the family $M(\theta)$ that gives the highest validation likelihood on B, since at optimum the shared model has to approximate the true underlying P_B well; 2) chose the family $T(x;\phi)$ that has fewer degrees of freedom then M, since otherwise the problem becomes underspecified. For example, consider M containing all univariate Gaussians parameterized by two parameters (μ, σ) aligned using polynomial transformations of the form $T(x; a_0, a_1, a_2) = a_2x^2 + a_1x + a_0$. In Example 3.2.1 we showed that Gaussian LRMF is equivalent to moment matching for two first moments, but with this choice of T, there exist infinitely many solutions for ϕ that all produce the desired mean and variance of the transformed dataset.

3.2.3 Experiments and Results

In this section, we present experiments that verify that minimizing the proposed LRMF objective (3.2.2) with Gaussian, RealNVP, and FFJORD density estimators indeed results in dataset alignment. We also show that both under- and over-parameterized LRMFs performed well in practice, and that resulting flows preserved the local structure of aligned datasets better than non-parametric objectives and the AlignFlow-inspired [39] baseline and were overall more stable than parametric adversarial objectives. We also show that the RealNVP LRMF produced a semantically meaningful alignment in the embedding space of an autoencoder trained simultaneously on two digit domains (MNIST and USPS) and preserved the manifold structure of one mesh surface distribution mapped to the surface distribution of a different mesh. We provide Jupyter notebooks with code in JAX [14] and TensorFlow Probability (TFP) [24].

Setup 1: Moons and blobs. We used LRMF with Gaussian, Real NVP, and FFJORD densities $P_M(x;\theta)$ with affine, NVP, and FFJORD transformations $T(x;\phi)$

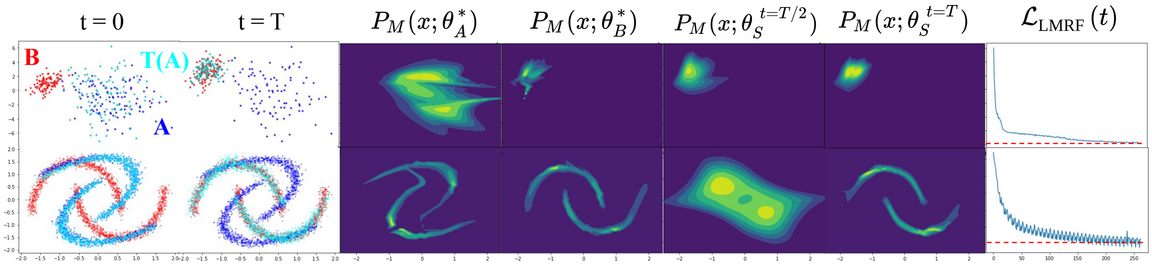


Figure 3·5: The dynamics of training a Real NVP LRMF on the blob (first row) and moons (second row) datasets. Blue, red and cyan points represent A, B and $T(A)$ respectively. The first two columns show $T(A)$ before and after training. The third and fourth columns show optimal models from M for A and B. The fifth and the sixth columns show the evolution of the shared model. The last column shows the LRMF loss over time. Even a severely *overparameterized* LRMF does a good job at aligning blob distributions. The animated version that shows the evolution of respective models is available on the project web page ai.bu.edu/lrmf. *Best viewed in color.*

respectively to align pairs of moon-shaped and blob-shaped datasets. The blobs dataset pair contains two samples of size $N = 100$ from two Gaussians. The moons dataset contains two pairs of moons rotated 50° relative to one another. We used original hyperparameters and network architectures from Real NVP [25] and FFJORD [36], the exact values are given in the supplementary. We also measured how well the learned LRMF transformation preserved the local structure of the input compared to other common minimization objectives (EMD, MMD) and the "back-to-back" composition of flows using a 1-nearest neighbor classifier trained on the target and evaluated on the transformed source. We also compared our objective to the adversarial network with spectral normalized discriminator (SN-GAN) in terms of how well their alignment quality can be judged based on the objective value alone.

Results. In agreement with Example 3.2.1, affine Gaussian LRMF matched the first two moments of aligned distributions. In Real NVP and FFJORD experiments, the shared model converged to θ_B^* gradually "enveloping" both domains and pushing them towards each other. In both under-parameterized (Gaussian LRMF on

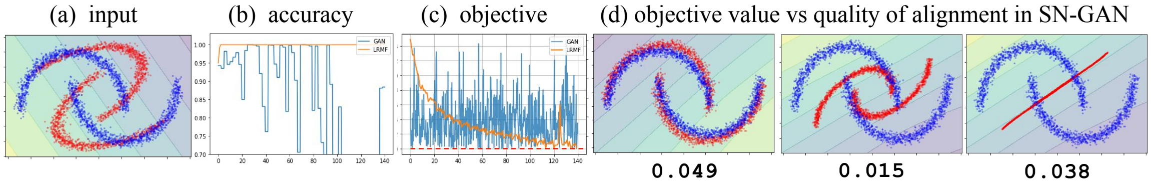

(a) input (b) accuracy (c) objective (d) objective value vs quality of alignment in SN-GAN

Figure 3·6: The dynamics of training a GAN with Spectral Normalization (SN-GAN) on the moons dataset. The adversarial framework provides means for aligning distributions against rich families of parametric discriminators but requires the right choice of learning rate and an external early stopping criterion because the absolute value of the adversarial objective (blue) is not indicative of the actual alignment quality even in low dimensions. The proposed LRMF method (orange) can be solved by plain minimization and converges to zero.

moons) and over-parameterized (RealNVP LRMF on blobs) regimes our loss successfully aligns distributions. In all experiments, the LRMF loss converged to zero in average (red line), so $\mathcal{E}(A, T, M) \approx 0$, meaning that affine, Real NVP, and FFJORD transformations keep input distributions “equally far“ from M. The loss occasionally dropped below zero because of the variance in mini-batches. Figure 3·7 shows that, despite good marginal alignment (top row) produced by MMD, EMD, and the $F \circ G^{-1}$ composition (inspired by AlignFlow [39]), the alignment produced by LRMF preserved the local structure of transformed distributions better, comparably to the SN-GAN both qualitatively (color gradients remain smooth in the middle row) and quantitatively in terms of adapted 1-NN classifier accuracy (bottom row). We believe that LRMF and SN-GAN preserved the local structure of presented datasets better than non-parametric models because assumptions about aligned distributions are too general in the non-parametric setting (overall smoothness, etc.), i.e. parametric models (flows, GANs) are better at capturing structured datasets. At the same time, Figure 3·6 shows that the quality of the LRMF alignment can be judged from the objective value (orange line) and stays at optima upon reaching it, while SN-GAN's performance (blue) can be hardly judged from the value of its adversarial objective

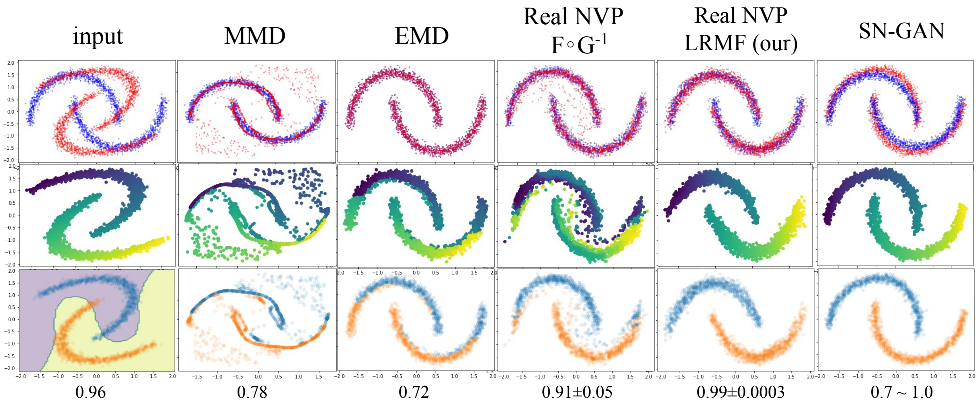
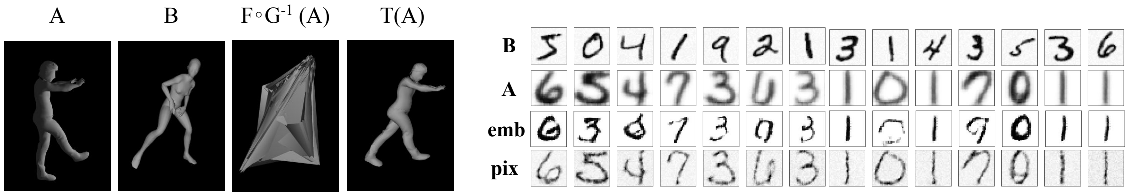


Figure 3·7: Among the non-adversarial alignment objectives, only LRMF preserves the manifold structure of the transformed dataset. Each domain contains two moons. The top row shows how well two domains (red and blue) are aligned by different methods trained to transform the red dataset to match the blue dataset. The middle row shows new positions of points colored consistently with the first column. The bottom row shows what happens to red moons after the alignment. Numbers at the bottom of each figure show the accuracy of the 1-nearest neighbor classifier trained on labels from the blue domain and evaluated on transformed samples from the red domain. The animated version is available on the project web page http://ai.bu.edu/lrmf.

and diverges even from near-optimal configurations.

Setup 2: Meshes. We treated vertices from two meshes as samples from two mesh surface point distributions and aligned them. After that, we draw faces of the original mesh at new vertex positions. We trained two different flows F and G on these surface distributions, passed one vertex cloud through their back-to-back composition, and compared this with the result obtained using LRMF.

Results. Figure 3·8a shows that, in agreement with the previous experiment, the number of points in each sub-volume of B matches the corresponding number in the transformed point cloud $F(G^{-1}(A))$, but drawing mesh faces reveals that the local structure of the original mesh surface manifold is distorted beyond recognition. The LRMF alignment (fourth column) better preserves the local structure of the original distribution - it rotated and stretched A to align the most dense regions (legs, torso,



(a) composition of flows on meshes vs LRMF

(b) A2B learned by LRMF in embedding space and pixel space

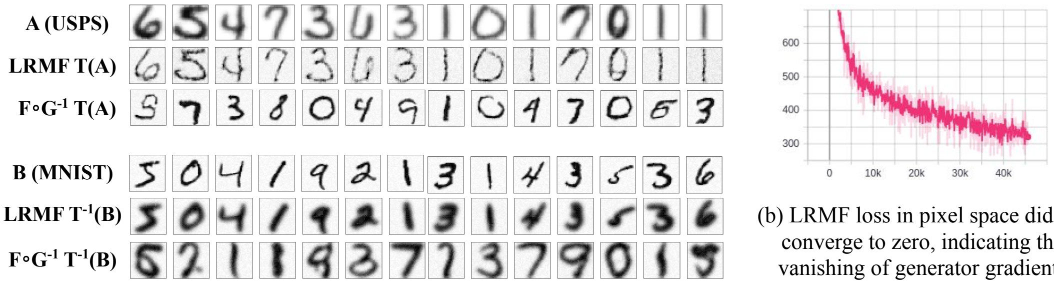
Figure 3·8: RealNVP LRMF successfully semantically aligned digits and preserved the local structure of the mesh surface manifold. **(a)** The marginal distribution produced by the "back-to-back" composition $F \circ G^{-1}$ of two normalizing flows trained on vertices of two meshes matches the point distribution of B, but the local structure of the original manifold is distorted, while LRMF preserves the local structure. **(b)** USPS digits (B) transformed into MNIST digits (A) via LRMF in VAE embedding space (emb), via LRMF in pixel space (pix).

head) with the most dense regions of B.

Setup 3: Digit embeddings. We trained a VAE-GAN to embed unlabeled images from USPS and MNIST into a shared 32-dimensional latent space. We trained a Real NVP LRMF to map latent codes of USPS digits to latent codes of MNIST. We also trained digit label classifiers on images obtained by decoding embeddings transformed using LRMF, CORAL, and EMD and applied the McNemar test of homogeneity [83] to the contingency tables of predictions made by these classifiers.

Results. The LRMF loss attained zero. Figure 3·8b(emb) shows that LRMF semantically aligned images form two domains. Classifiers trained on images transformed using LMRF had higher accuracy on the target dataset (.55 for LRMF vs .47 for EMD vs .48 for CORAL). McNemar test showed that LRMF's improvements in accuracy were significant (p-value $\ll$ 1e-3 in all cases).

Setup 4: GLOW. We trained a GLOW LRMF to align USPS and MNIST in 32x32 pixel space, visualized outputs of the forward and backward transformation, and the LRMF loss value over training iterations.

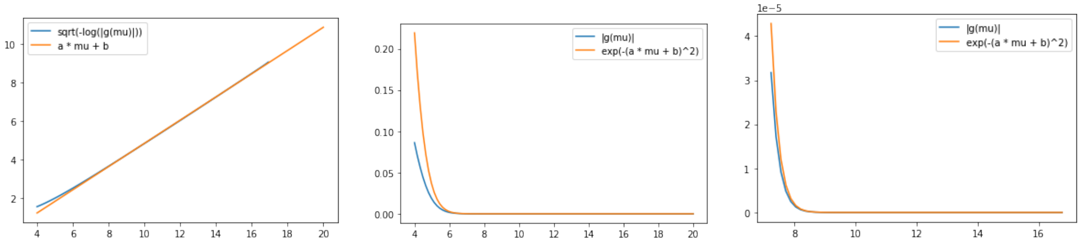


(b) LRMF loss in pixel space did not converge to zero, indicating the vanishing of generator gradients

(a) Glow-LRMF and the Align-Flow [12] inspired baseline in pixel space

Figure 3·9: GLOW-LRMF did not converge in pixel space, but preserved class labels much better than the AlignFlow-inspired baseline [39]. **(a)** Images generated by applying learned flow models in forward and backward directions to USPS and MNIST respectively. **(b)** GLOW-LRMF loss did not converge to zero due to the vanishing gradients in higher dimensions (pixel space). This failure mode can be detected by looking at the loss values alone.

Results. The model learned to match the stroke width across domains but did not make images completely indistinguishable (Figure 3·9). The shared density model converged to the local minima that corresponds to approximating $T(A)$ and B as two distinct "bubbles" of density that fail to merge. This is the same failure mode we illustrated in Example 3.2.2 where two components of the shared model get stuck approximating datasets that are too far away and fail to bring the model into the deeper global minima. We would like to note that even though the loss did not converge to zero, i.e. the model failed to find a marginally perfect alignment, it did so *not silently*, in stark contrast with adversarial methods that typically fail silently. These results agree with our hypothesis about vanishing transformation gradients in higher dimensions (end of Section 2), resulting in vast flat regions in the LRMF loss landscape with respect to the transformation parameter ϕ, obstructing full marginal alignment. The AligFlow-inspired [39] composition of flows ($F \circ G^{-1}$ in Figure 3·9), on the other hand, produced very good marginal alignment, judging from the fact that transformed images look very much like MNIST and USPS digits, but erased



(a) **(b)** **(c)**

Figure 3·10: Gradient of the cross-entropy of between two mixture models as a function of the mean of one of the first components of the first mixture to illustrate the Example 3.2.2, estimated using JAX. **(a)** $\sqrt{-\log(|\partial L/\partial\mu|)}$ vs $a\mu+b$; **(b)** $|\partial L/\partial\mu|$ vs $\exp(-(a\mu+b)^2)$; **(c)** same as in the middle pane, but for $\mu\in[7,16]$.

the semantics in the process, judging from the mismatch between classes of original and transformed digits.

Vanishing Gradient Simulation. We approximated $|\partial H[m_1, m_2(\mu)]/\partial\mu|$, where m_1 and $m_2(\mu)$ are two equal mixtures of normal distributions, by computing the partial derivative using auto-differentiation in JAX. The objective was $L = \text{logsumexp}(\{\log(p_i(X;\mu)) + \log 2\}_i)$, where $\log p_i(x;\mu)$ is a log probability of the mixture component from m_2, and X is a fixed large enough (n=100k) sample from the m_1. Figure 3·10 shows that $\sqrt{-\log(|\partial L/\partial\mu|)}$ fits to $a\mu+b$ for $a=0.6, b=-1.168$ with $R=0.99996$, therefore making us believe that $\|[\partial L(b,\mu)/\partial\mu](0,\mu)\| \propto \exp(-\mu^2)$.

Conclusion. To sum up, in this section we propose a new alignment objective parameterized by a deep density model and a normalizing flow that, when converges to zero, guarantees that the density model fitted to the transformed source dataset is optimal for the target and vice versa. We also show that the resulting model is robust to model misspecification and preserves the local structure better than other non-adversarial objectives. We showed that minimizing the proposed objective is equivalent to training a particular GAN, but is not subject to mode collapse and instability of adversarial training, however in higher dimensions, is still affected by the vanishing of generator gradients. Translating recent advances in dealing with the vanishing of generator gradients, such as instance noise regularization [2, 100, 107], to the language of likelihood-ratio minimizing flows offers an interesting challenge for future research.

3.3 Improving Semantic Consistency using Honesty Losses

Author Contribution. Findings concerning self-adversarial defense techniques described in this section were first reported by Bashkirova et al. [7]. Ben Usman helped constructing datasets, automating evaluation, aggregating results across baselines, and motivating the problem, but the core technical contribution of the proposed self-adversarial defense methods should be attributed to its first author.

While stabilizing the alignment objective using methods described in the first half of this chapter results in better convergence, the learned mapping might still be nonsensical. In this section, we show that an adversarial attack that takes place in such models results in poor semantic consistency of the learned domain alignment mapping, and propose several solutions to this issue.

Unsupervised image-to-image translation methods [71, 127] can infer semantically meaningful one-to-one cross-domain mappings from pairs of semantically related sets of images (domains) without pair supervision. In this section, we show all models that use the cycle-consistency loss (2.13), including the original CycleGAN [127], learn to reconstruct input images by embedding low-amplitude structured noise into intermediate generated images. We propose an adversarial loss that prevents this "cheating" and, as a result, improves translation accuracy.

As discussed previously, suppose we are given independent samples from two image domains A and B, and our goal is to learn mappings from one visual domain into the other. In line with the notation introduced in the original CycleGAN [127] paper, we train two cross-domain mappings $G : A \to B$ and $F : B \to A$ to generate plausible examples of respective domains by pitting them against two discriminators D_A and D_B trained to classify whether the input image is a true representative of the corresponding domain or was generated by G or F accordingly. These mappings are regularized using the pixel-wise cycle-consistency loss between the input image and its cycle reconstruction:

$$\mathcal{L}_{rec} = \mathbb{E}_{x \sim A} \ \left\| F(G(x)) - x \right\|_1 + \mathbb{E}_{x \sim B} \ \left\| G(F(x)) - x \right\|_1 \tag{3.31}$$

However, in the case when domain A is richer than B, the mapping G is many-to-one (i.e. for each image in B there are multiple correct correspondences in A), the generator is still forced to perfectly reconstruct the input even though some of the information of the input image is lost after the translation to the domain B. As shown in [21], such behavior of a CycleGAN can be described as an adversarial attack, and in fact, for any given image it is possible to generate such structured noise that would lead to a perfect reconstruction of the target image.

In this section, we show that methods that use cycle-consistency loss (3.31) add a low-amplitude signal to the translation $\hat{y}$ that is invisible to the human eye. The addition of this signal is enough to reconstruct the information of image x that should not be present in $\hat{y}$. This makes methods that incorporate cycle-consistency loss sensitive to low-amplitude high-frequency noise since that noise can destroy the hidden signal (shown in Figure 3·11). In addition, such behavior can force the model to converge to a non-optimal solution or even diverge since by adding structured noise the model "cheats" to minimize the reconstruction loss instead of learning the correct mapping.

3.3.1 Honest CycleGAN

In this subsection, we introduce two defense techniques that prevent cycle-consistent models from embedding such adversarial signals into generated images.

Adversarial training with noise. One approach to defending the model from a self-adversarial attack is to train it to be resistant to the perturbation of nature similar to the one produced by the hidden embedding. Unfortunately, it is impossible to separate the pure structured noise from the translated image, so classic adversarial defense training cannot be used in this scenario. However, it is possible to prevent the model from learning to embed by adding perturbations to the translated image before reconstruction. The intuition behind this approach is that adding random noise of amplitude similar to the hidden signal disturbs the embedded message. This results in a high reconstruction error, so the generator cannot rely on the embedding. The modified noisy cycle-consistency loss can be described as follows:

$$\mathcal{L}_{rec}^{noisy} = \left\| F(G(x) + \varepsilon(\theta_n)) - x \right\|_1 , \tag{3.32}$$

where $\varepsilon(\theta_n)$ is some high-frequency perturbation function with parameters θ_n.

Guess Discriminator. Ideally, the self-adversarial attack should be detected by the discriminator, but this might be too hard for it since it never sees real and fake examples of the same content. In the supervised setting, this problem is naturally solved by conditioning the outputs on the ground truth labels. For example, a self-adversarial attack does not occur in Conditional GANs, such as pix2pix [49], because the discriminator is conditioned on the ground truth class labels and is provided with real and fake examples of each class. In the unsupervised setting, however, there is no such information about the class labels, and the discriminator only receives unpaired real and fake examples from the domain. This task is significantly harder for the discriminator as it has to learn the distribution of the whole domain. One widely used defense strategy is adding adversarial examples to the training set. While it is possible to model the adversarial attack of the generator, it is very time and memory consuming as it requires training an additional network that generates such examples at each step of training the GAN. However, we can use the fact that cycle-consistency loss forces the model to minimize the difference between the input and reconstructed images, so we can use the reconstruction output to provide the fake example for the real input image as an approximation of the adversarial example.

Method	**ACC** ↑	**IoU**↑	**IoU p2p**↑	**RH**↓	**SN**↓
CycleGAN	0.23	0.16	0.20	27.43 ± 6.1	446.9
CycleGAN + noise*	**0.24**	0.17	0.23	9.17 ± 7.4	**94.2**
CycleGAN + guess*	0.24	**0.17**	0.21	11.4 ± 7.0	212.6
CycleGAN + guess + noise*	0.236	**0.17**	**0.24**	**6.1** ± **5.9**	150.6
UNIT	0.08	0.04	0.06	6.4 ± 11.7	361.5
MUNIT + cycle	0.13	0.08	0.17	2.5 ± 8.9	244.9
pix2pix (supervised)	0.4	0.34	–	–	–

Table 3.1: Results on the GTA V dataset. *acc. segm* and *IoU segm* represent mean class-wise segmentation accuracy and IoU, *IoU p2p* is the mean IoU of the pix2pix segmentation of the segmentation-to-frame mapping; *RH* (Eq.3.34) and *SN*(Eq.3.35) are the quantized reconstruction honesty and sensitivity to noise of the many-to-one mapping (B2A2B) respectively. * – our proposed defense methods. The reconstruction error distributions plots can be found in the supplementary material (Section 2).

Thus, the defense during training can be formulated in terms of an additional *guess discriminator* that is very similar to the original GAN discriminator, but receives as input two images – input and reconstruction – in a random order, and "guesses" which of the images is fake. As with the original discriminator, the guess discriminator D_{guess} is trained to minimize its error while the generator aims to produce such images that maximize it. The guess discriminator loss or *guess loss* for domain A with guess discriminator D^A_{guess} can be written down as:

$$\mathcal{L}^A_{guess} = \phi(D^A_{guess}(X, F(G(X))) + \phi(1 - D^A_{guess}(F(G(X)), X)) \tag{3.33}$$

where $X \sim P_A$, $D^A_{guess}(X, \hat{X}) \in [0, 1]$ is the predicted probability that X is real and $\hat{X}$ is reconstruction, and $\phi(x)$ is a discriminator loss function ($\phi(x) = \log(x)$ for the original GAN, and $\phi(x) = x^2$ for LSGAN we used). This loss resembles the class label conditioning in the Conditional GAN in the sense that the guess discriminator receives real and fake examples that are presumably of the same content, therefore the embedding detection task is significantly simplified.

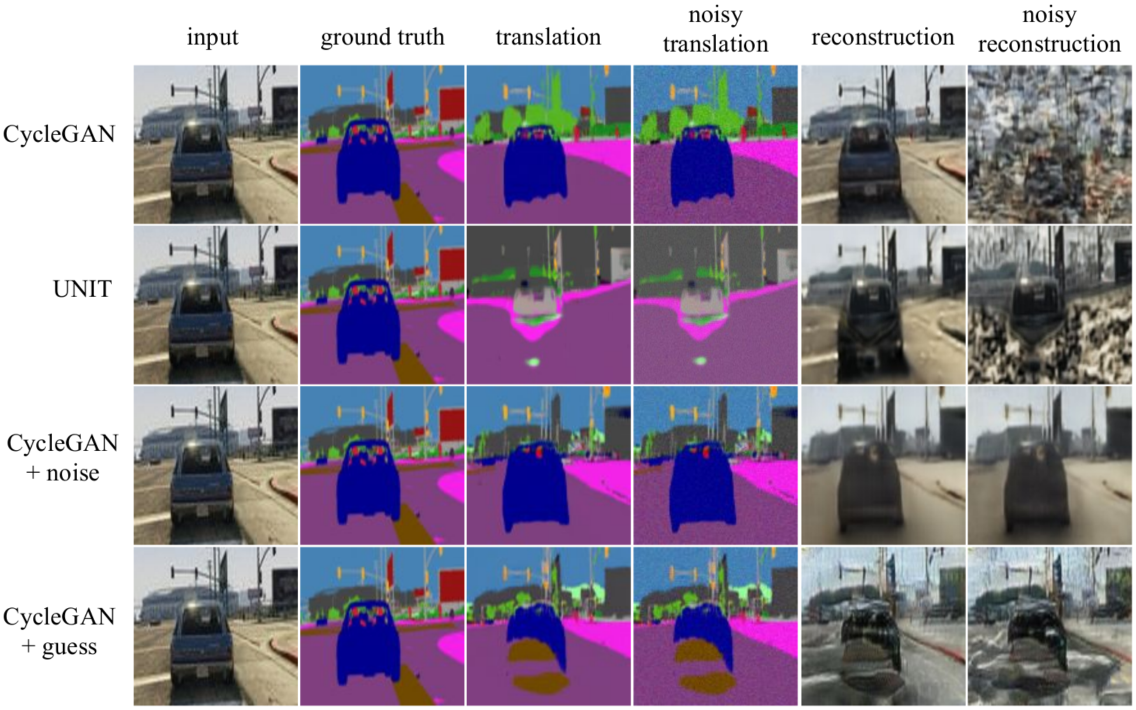


Figure 3·11: Results of translation of GTA [99] frames to semantic segmentation maps using CycleGAN, UNIT, and CycleGAN with our two proposed defense methods, additive noise and guess loss. The last column shows the reconstruction of the input image when high-frequency noise (Gaussian noise with mean 0 and standard deviation 0.08 ∼ 10 intensity levels out of 256) is added to the output map. Ideally, if the reconstruction is "honest" and relies solely on the visual features of the input, the reconstruction quality should not be greater than that of the translation. The results of all three translation methods (CycleGAN, UNIT, and MUNIT) show that the reconstruction is almost perfect regardless of the translation accuracy. Furthermore, the reconstruction of the input image is highly sensitive to low-amplitude random noise added to the translation. Both of the proposed self-adversarial defense techniques (Section 3.3) make the CycleGAN model more robust to the random noise and make it rely more on the translation result rather than the adversarial structured noise as in the original CycleGAN and UNIT. More translation examples can be found in Section 3 of the supplementary material in the original paper [7]. *Best viewed in color.*

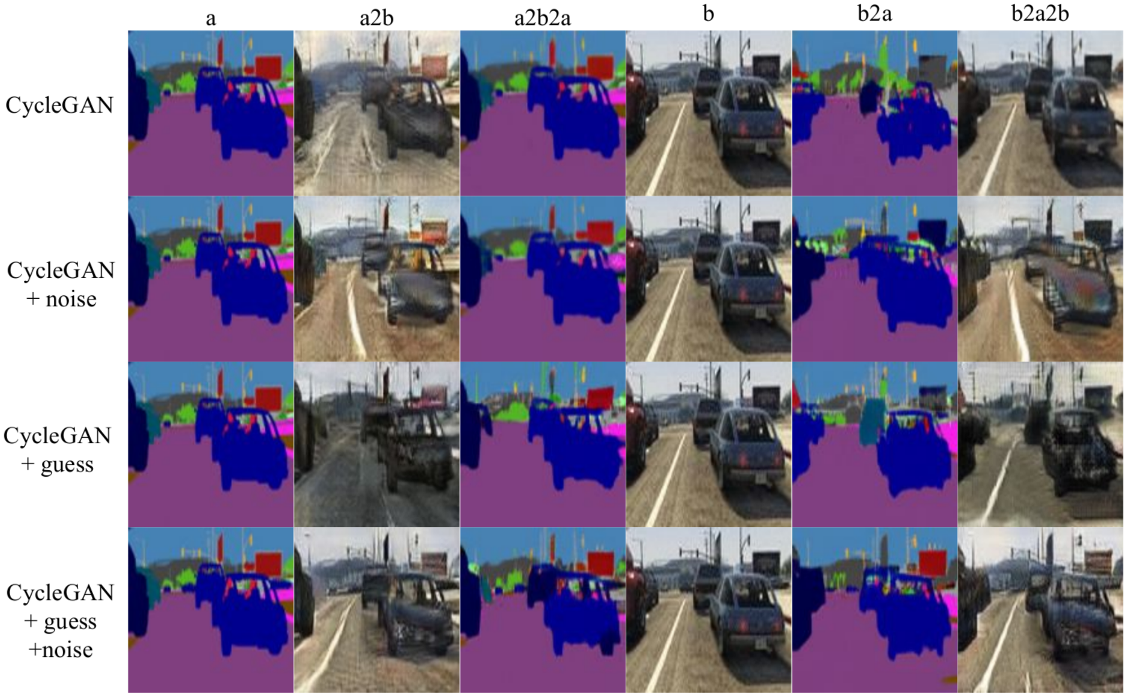


Figure 3·12: Results of the GTA frames-to-segmentation translation with the original CycleGAN and our defense techniques. The frame reconstruction (b2a2b) with noisy CycleGAN is remarkably similar to the opposite translation (a2b). For example, the road marking in the reconstructed image is located at the same place as in the translation (a2b) rather than as in the input (b).

3.3.2 Experiments and Results

In the abundance of GAN-based methods for unsupervised image translation, we limited our analysis to three popular state-of-art models that cover both unimodal and multimodal translation cases: CycleGAN [127], UNIT [71], and MUNIT [47]. To provide empirical evidence of our claims, we performed a sequence of experiments on three publicly available image-to-image translation datasets. Despite the fact that all three datasets are paired and hence the ground truth correspondence is known, the models that we used are not capable of using the ground-truth alignment by design and thus were trained in an unsupervised manner.

Playing for Data (GTA) Dataset. Out split of the original dataset [99] consists of 24966 pairs of image frames and their semantic segmentation maps. We used a subset of 10000 frames (7500 images for training, 2500 images for testing) with day-time lighting resized to 192 × 192 pixels, and randomly cropped with window size 128 × 128.

Translation quality metric. The choice of aligned datasets was dictated by the need to quantitatively evaluate the translation quality which is impossible when the ground truth correspondence is unknown. However, even having the ground truth pairs does not solve the issue of quality evaluation in one-to-many case, since for one input image there exist a large (possibly infinite) number of correct translations, so pixel-wise comparison of the ground truth image and the output of the model does not provide a correct metric for the translation quality. In order to overcome this issue, we adopted the idea behind the Inception Score [105] and trained the supervised Pix2pix [49] model to perform many-to-one mapping as an intermediate step in the evaluation. Considering the GTA dataset example, in order to evaluate the unsupervised mapping from segmentation maps to real frames (later on – segmentation to real), we train the Pix2pix model to translate from real to segmentation; then we feed it the output of the unsupervised model to perform "honest" reconstruction of the input segmentation map, and compute the Intersection over Union (IoU) and mean class-wise accuracy of the output of Pix2Pix when given a ground truth example and the output of the one-to-many translation model. For any ground truth pair (A_i, B_i), the one-to-many translation quality is computed as $\mathrm{IoU}(pix(G_A(B_i)), pix(A_i))$, where $pix(\cdot)$ is the translation with Pix2pix from A to B. The "honest reconstruction" is compared with the Pix2pix translation of the ground truth image A_i instead of the ground truth image itself in order to take into account the error produced by the Pix2pix translation.

Reconstruction honesty metric. Since it is impossible to acquire the structured noise produced as a result of a self-adversarial attack, there is no direct way to either detect the attack or measure the amount of information hidden in the embedding. In order to evaluate the presence of a self-adversarial attack, we developed a metric that we call *quantized reconstruction honesty*. The intuition behind this metric is that, ideally, the reconstruction error of the image of the richer domain should be the same as the one-to-many translation error if given the same input image from the poorer domain. In order to measure whether the model is independent of the origin of the input image, we quantize the many-to-one translation results in such a way that it only contains the colors from the domain-specific palette. In our experiments, we approximate the quantized maps by replacing the colors of each pixel with the closest one from the palette. We then feed those quantized images to the model to acquire the "honest" reconstruction error and compare it with the reconstruction error without quantization. The honesty metric for a one-to-many reconstruction can be described as follows:

$$RH = \frac{1}{N}\sum_{i=1}^{N}\{\|G_A(\lfloor G_B(X_i)\rfloor) - Y_i\|_2 - \|G_A(G_B(X_i)) - Y_i\|_2\}, \tag{3.34}$$

where $\lfloor * \rceil$ is a quantization operation, G_B is a many-to-one mapping, (X_i, Y_i) is a ground truth pair of examples from domains A and B.

Sensitivity to noise. Aside from the obvious consequences of the self-adversarial attack, such as convergence of the generator to a suboptimal solution, there is one more significant side effect of it – extreme sensitivity to perturbations. Figure 3·11 shows how the addition of low-amplitude Gaussian noise effectively destroys the hidden embedding thus making a model that uses cycle-consistency loss unable to correctly reconstruct the input image. In order to estimate the sensitivity of the model, we add zero-mean Gaussian noise to the translation result before reconstruction and compute the reconstruction error. The sensitivity to noise of amplitude σ for a set of images $X_i \sim p_A$ is computed by the following formula:

$$SN(\sigma) = \frac{1}{N}\sum_{i=1}^{N} \|G_A(G_B(X_i) + \mathcal{N}(0,\sigma)) - G_A(G_B(X_i))\|_2 \qquad (3.35)$$

The overall sensitivity of a method is computed as an area under the curve:

$$AuC(SN(\sigma)) = \int_{a}^{b} SN(x)dx \qquad (3.36)$$

In our experiments we chose $a = 0$, $b = 0.2$, $N = 500$ for Google Maps and GTA experiments and $N = 100$ for the SynAction experiment. In case when there is no structured noise in the translation, the reconstruction error should be proportional to the amplitude of added noise, which is what we observe for the one-to-many mapping using MUNIT and CycleGAN. Surprisingly, UNIT translation is highly sensitive to noise even in the one-to-many case.

The many-to-one mapping result (Figure 3·11), in contrast, suggests that the structured noise is present since the reconstruction error increases rapidly and quickly saturates at noise amplitude 0.08. The results of one-to-many and many-to-one noisy reconstruction show that both noisy CycleGAN and guess loss defense approaches make the CycleGAN model more robust to high-frequency perturbations compared to the original CycleGAN.

Results. The results of our experiments show that the problem of self-adversarial attacks is present in all three cycle-consistent methods we examined. The noise-regularization defense helps the CycleGAN model to become more robust both to small perturbations and to the self-adversarial attack. The guess loss approach, on the other hand, while allowing the model to hide some small portion of information about the input image (for example, road marking for the GTA experiment), produces more interpretable and reliable reconstructions. Furthermore, the combination of both proposed defense techniques results beats both methods in terms of translation quality and reconstruction honesty (Figure 3·12). Since both defense techniques force the generators to rely more on the input image than on the structured noise, their results are more interpretable and provide a deeper understanding of their "reasoning". For example, since the training set did not contain any examples of a truck that is colored in white and green, at test time the guess-loss CycleGAN approximated the green part of the truck with the "vegetation" class color and the white part with the building class color (see Section 3 of the supplementary material); the reconstructed frame looked like a rough approximation of the truck despite the fact that the semantic segmentation map was wrong. This can give a hint about the limitations of the given training set.

Chapter 4

Cross-Domain Image Manipulation

In the previous chapter, we showed how to improve the stability and semantic consistency of adversarial alignment. In this chapter, we show how to use these adversarial alignment methods to manipulate individual factors of real images using cross-domain supervision. In this section, we show that the adversarial alignment alone is not enough to efficiently manipulate individual factors of real images and propose novel components that enable such controlled manipulation. For example, Figure 4·1a-b shows what happens if we apply an alignment method, such as CycleGAN [21], to a pair of domains where the first (real) domain is our application domain that lacks any supervision, and the second (synthetic) domain provides fine control over all factors of variation. The learned mapping would not be very useful for image manipulation, since mapping an image to the synthetic domain, manipulating it there, and mapping it back, would randomly alter all aspects of the input image not reflected in the simulation. In Section 4.1 we investigate how we can use adversarial domain alignment to transfer fine control over individual factors present in the simulation onto a real domain while preserving all other aspects of the input image intact. While we, of course, can not use synthetic supervision to learn to manipulate individual factors of real images absent from the synthetic domain, as illustrated in Figure 4·1c, in Section 4.2 we investigate how adversarial domain alignment can help us differentiate factors of variation shared across both domains from those unique to each domain, and manipulate factors in these two groups independently from each other.

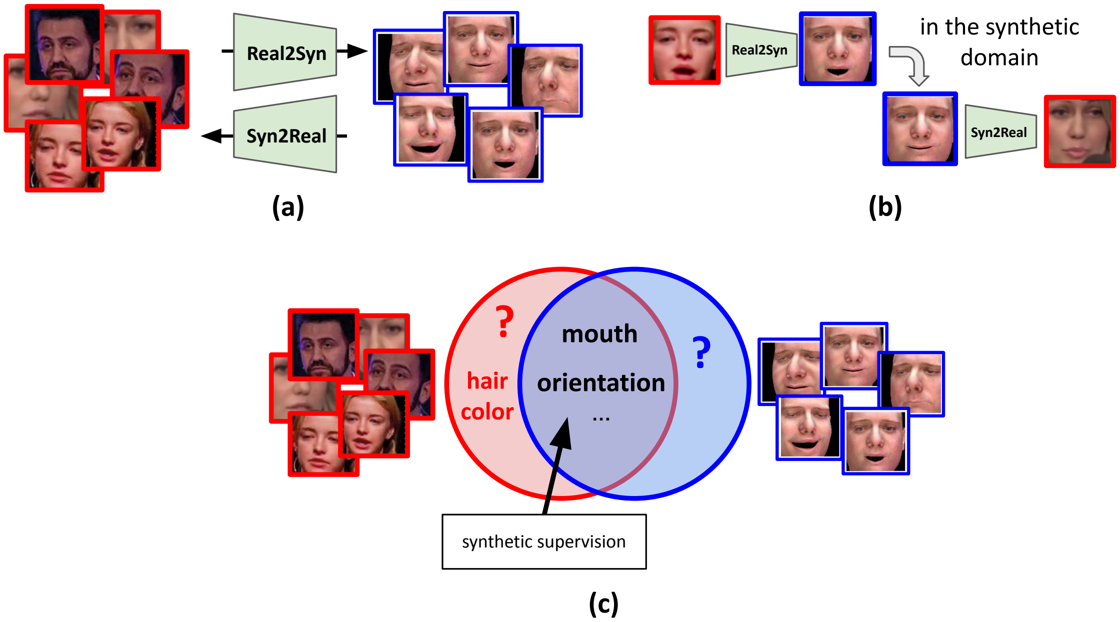


Figure 4·1: Challenges in performing controlled cross-domain image manipulation via domain alignment. **(a)** While an unsupervised adversarial alignment method can learn a pair of mappings between uncontrolled (real) and controlled (synthetic) datasets, **(b)** translating a real image into the synthetic domain, applying a transformation there, and mapping it back to the real domain is not practical, since it changes many other attributes of the input image (e.g. identity in this example). In Section 4.1 we show how to overcome these challenges and train a model that can manipulate individual factors of real images using synthetic supervision. **(c)** While synthetic supervision is not available for factors absent from the synthetic domain (such as hair color in this example), in Section 4.2 we show how one to learn to differentiate factors present in both domains from domain-specific ones without any pair supervision and manipulate these two groups of factors independently from each other.

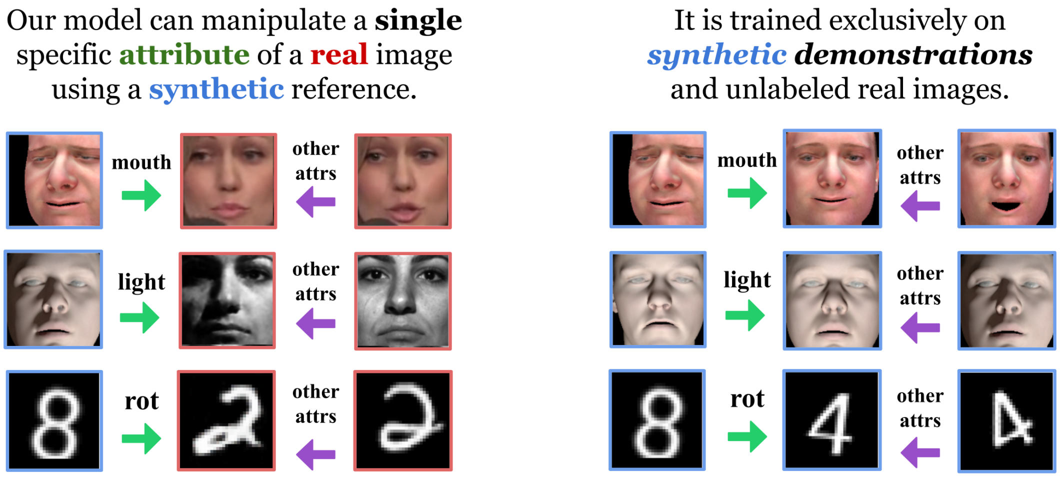

Our model can manipulate a **single** specific **attribute** of a **real** image using a **synthetic** reference.
It is trained exclusively on ***synthetic demonstrations*** and unlabeled real images.
mouth
other attrs
light
other attrs
rot
other attrs
mouth
other attrs
light
other attrs
rot
other attrs

Figure 4·2: The task of image manipulation from cross-domain demonstrations. The model has to learn to manipulate attributes of real images using only synthetic supervision.

4.1 Image Manipulation via Cross-Domain Demonstrations

In this section, we propose PuppetGAN - a deep model for targeted and controlled modification of natural images that requires neither explicit attribute labels nor a precise simulation of the real domain. To enable control over a specific attribute in real images PuppetGAN only requires "demonstrations" of how the desired attribute manipulation affects the output of a crude simulation, see Figure 4·2. Our method uses these synthetic demonstrations to supervise attribute disentanglement in the synthetic domain and extends this disentanglement to the real domain. We quantitatively evaluate how well our model can preserve other attributes of the input when a single attribute is manipulated. To sum up, in this section we:

1. Introduce a new challenging "cross-domain image manipulation by demonstration" task: the model has to manipulate a specific attribute of a real image to match a synthetic reference image using *only* examples of real images and

attribute labels for	**single-domain**	**cross-domain**
single domain	Mathieu *et al.* [82], Cycle-VAE [41], Szabó *et al.* [111]	E-CDRD [73], DiDA [15], **PuppetGAN**
both domains	—	UFDN [70]
unsupervised	InfoGAN [18], β-VAE [43], β-TCVAE [17]	DRIT [62], MUNIT [47]

Table 4.1: Some of existing disentanglement methods that enable controlled manipulation of real images.

demonstrations of the desired attribute manipulation in *the synthetic domain* at train time in the presence of a significant domain shift both in the domain appearance and attribute distributions.

2. Propose a model that enables controlled manipulation of a specific attribute and correctly preserves other attributes of the real input. We are the first to propose a model that enables this level of control under these data constraints.

3. Propose both proof-of-concept (digits) and realistic (faces and face renders) dataset pairs and a set of metrics for this task. We are the first to quantitatively evaluate the effects of cross-domain disentanglement on values of other (non-manipulated) attributes of images.

4.1.1 Background

Parametric domain models. Following recent advances in differentiable graphics pipelines [78], and high-quality morphable models [95], the work of Thies et al. [114] proposed a way to perform photo-realistic face expression manipulation and reenactment that cannot be reliably detected even by trained individuals. Unfortunately, methods like these rely on precise parametric models of the target domain and accurate fitting of these parametric models to input data. These most. Then, in order to manipulate a single property of an input image, all other properties (such as head pose, lighting, and facial expression in case of face manipulation) have to be estimated from an image and passed to a generative model together with a modified attribute to essentially "rerender" a new image from scratch. This approach enables visually superb image manipulation, but requires a detailed domain model capable of precisely modeling all aspects of the domain and re-rendering any input image from a vector of its attributes - it is a challenging task, and its solution often does not generalize to other domains.

Fully-supervised methods. An overview of generative neural models with full supervision is beyond the scope of this thesis, but it would include attribute supervision [20], supervision with scene graphs and segmentation [3], etc.

Single-Domain Disentanglement. One alternative to full domain simulation is learning a representation of the domain in which the property of interest and other properties could be manipulated independently - a so-called "disentangled representation". We summarized several kinds of disentanglement methods that enable such control over real images using simulated examples in Table 4.1. Supervised single-domain disentanglement methods require either explicit or weak (pairwise similarity) labels [41, 82, 111] for real images - a much stronger data requirement than the one we consider. As discussed in the seminal work of Mathieu *et al.* [82] on disentangling representations using adversarial learning and partial attribute labels and later explored in more detail by Szabó *et al.* [111] and Harsh Jha *et al.* [41], there are always degenerate solutions that satisfy the proposed constraints but cheat by ignoring one component of the embedding and hiding information in the other, we discuss steps we undertook to combat these solutions in the model and experiment sections.

Supervised Cross-Domain Disentanglement. In the presence of the second domain, one intuitive way of addressing the visual discrepancy between the two is to treat the domain label as just another attribute [70] and perform disentanglement on the resulting single large partially labeled domain. This approach enables interpolation between domains, and training conditional generative models using labels from a single domain, but does not provide means for manipulation of existing images across domains unless explicit labels in both domains are provided. Recent papers [15, 73] suggested using explicit categorical labels to train explicit attribute classifiers on the synthetic domain and adapt it to the real domain; the resulting classifier is used to (either jointly or in stages) disentangle embeddings of real images. These works showed promising results in manipulating categorical attributes of images to augment existing datasets (like face attributes in CelebA [74] or class labels in MNIST), but neither of these methods was specifically designed for or tested for their ability to *preserve other attributes of an image*: if we disentangle the size of a digit from its class for the purpose of, effectively, generating more target training samples for classification, we do not care whether the size is preserved when we manipulate the digit class since that would still yield a correctly "pseudo-labeled" sample from the real domain. Therefore, high classification accuracies of adapted attribute classifiers (reported in these papers) do not guarantee the quality of disentanglement and the ability of these models to preserve unlabeled attributes of the input. Moreover, these methods require explicit labels making them not applicable to a wide range of attributes that are hard to express as categorical labels (shape, texture, lighting). In this work, we specifically focus on manipulating individual attributes of images using demonstrations from another domain, in the presence of a significant domain shift (both visual and in terms of distributions of attribute values) and explicitly quantitatively evaluate the ability of our model to preserve all attributes other the one we

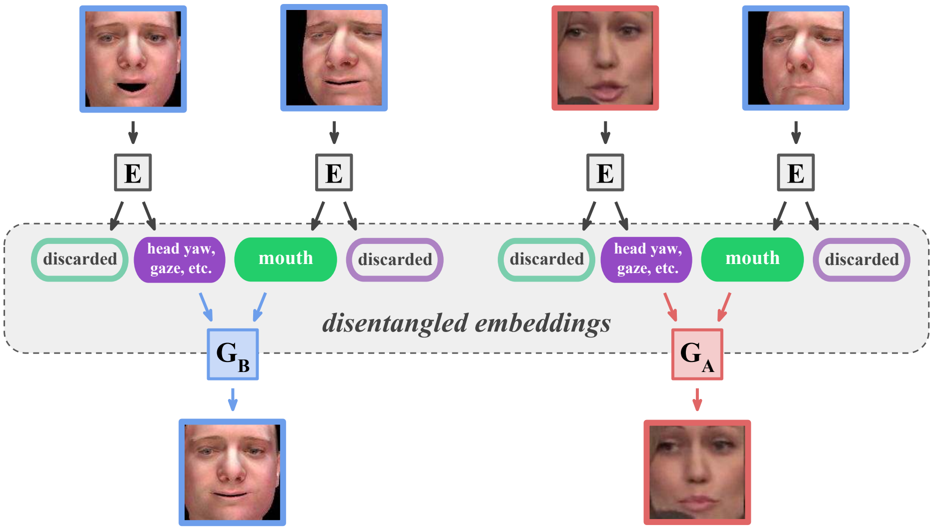


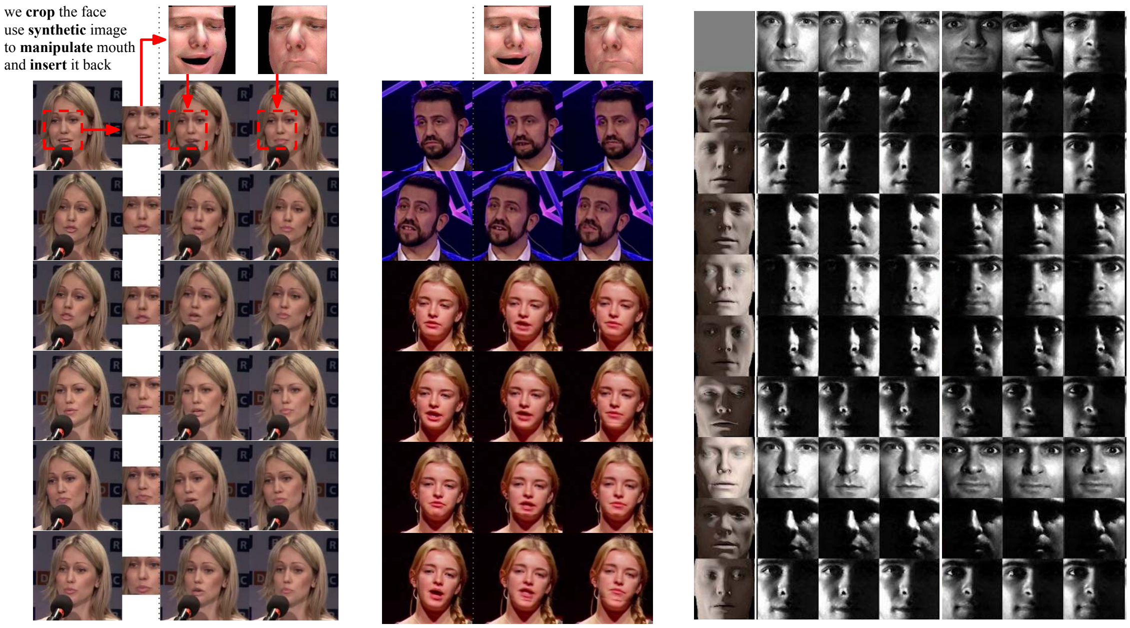
Figure 4·3: PuppetGAN overview: we train a domain-agnostic encoder (E), a decoder for the real domain (G_A), and a decoder for the synthetic domain (G_B) to disentangle the attribute we would like to control in real images (the "attribute of interest" - AoI - mouth expression in this example), and all other attributes (head orientation, gaze direction, microphone position in this example) that are not labeled or even not present (*e.g.* microphone) in the synthetic domain. Our model is trained on demonstrations of how the AoI is manipulated in synthetic images and individual examples of real images. At *test time*, a real image can be manipulated with a synthetic reference input by applying a real decoder to the attribute embedding of the reference image (green capsule) combined with the remaining embedding part (purple capsule) of the real input.

manipulated.

4.1.2 PuppetGAN

In this subsection, we formally introduce our data constraints, define a disentangled encoder and domain decoders used in the loss, and describe a set of constraints that ensure proper disentanglement of synthetic images and extension of this disentanglement to a real domain.

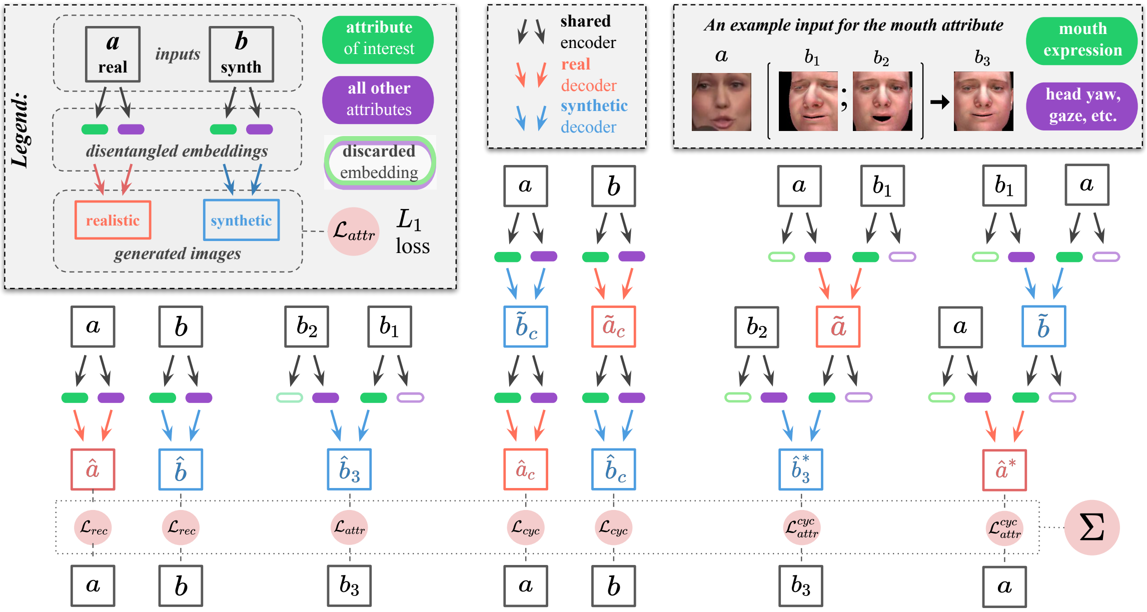
Setup. Consider having access to individual real images $a \in \mathcal{X}_A$, and triplets of synthetic images $(b_1, b_2, b_3) \in \mathcal{X}_B$ such that (b_1, b_3) share the attribute of interest



(a) mouth manipulation in 300-VW

(b) relighting faces from YaleB

Figure 4·4: More examples with other identities are provided in the supplementary. (a) When trained on face crops from a single 300-VW [103] video, PuppetGAN learns to manipulate mouth expression while preserving head orientation, gaze orientation, expression, *etc.* so well that directly "pasting" the manipulated image crop back into the frame without any stitching yields realistically manipulated images without noticeable head orientation or lighting artifacts (chin stitching artifacts area are unavoidable unless an external stitching algorithm is used); the video demonstration is available in the supplementary and at `http://bit.ly/iccv19_pupgan`. (b) When trained on face crops of all subjects from YaleB [33] combined into a single domain, PuppetGAN learns to properly apply lighting (AoI) from a synthetic reference image and correctly preserves subjects' identities without any identity labels; lighting of the original real image has little to no effect on the output.



(a) reconstruction loss (b) disentanglement loss (c) cycle loss (d) attribute cycle loss

Figure 4·5: Supervised losses jointly optimized during the training of the PuppetGAN. When combined, these losses ensure that the "attribute embedding" (green capsule) affects only the attribute of interest (AoI) in generated images and that the "rest embedding" (purple capsule) does not affect the AoI in generated images. When trained, manipulation of AoI in real images can be performed by replacing their attribute embedding components. Unsupervised (GAN) losses are not shown in this picture. An example at the top right corner illustrates sample images fed into the network to disentangle mouth expression (AoI) from other face attributes in real faces. Section 4.1.2 provides more details on the intuition behind these losses.

(AoI - the attribute that we want to control in real images), whereas the pair (b_2, b_3) shares all other attributes present in the synthetic domain. See the top right corner of Figure 4·5 for an example of inputs fed into the network to learn to control mouth expression (AoI) in real faces using crude face renders.

Model. The learned image representation consists of two real-valued vectors e_{attr} and e_{rest} denoted as green and purple capsules in Figures 4·3 and 4·5. We introduce domain-agnostic encoders for the attribute of interest E_{attr} and all other attributes E_{rest}, and two domain-specific decoders G_A, G_B for the real and synthetic domains respectively:

$$\begin{aligned} E_{attr} : (x) \mapsto e_{attr}, \quad & G_A : (e_{attr}, e_{rest}) \mapsto x_a \\ E_{rest} : (x) \mapsto e_{rest}, \quad & G_B : (e_{attr}, e_{rest}) \mapsto x_b. \end{aligned}$$

To simplify the loss definitions below, we introduce the domain-specific "attribute combination operator" that takes a pair of images (x, y), each from either of two domains, combines embeddings of these images, and decodes them as an image in the specified domain K:

$$C_K(x, y) \triangleq G_K\big(E_{attr}(x), E_{rest}(y)\big), \;\; K \in \{A, B\}.$$

Losses. We would like $C_K(x, y)$ to have the AoI of x and all other attributes of y, but we can not enforce this directly as we did not introduce any explicit labels. Instead, we *jointly* minimize the weighted sum of L_1-penalties for violating the following constraints illustrated in Figure 4·5 with respect to all parameters of both encoders and decoders:

(a) the reconstruction constraint ensures that encoder-decoder pairs actually learn representations of respective domains

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{x \sim \mathcal{X}_A} \; ||x - C_A(x, x)|| + \mathbb{E}_{x \sim \mathcal{X}_B} \; ||x - C_B(x, x)|| \tag{4.1}$$

(b) the disentanglement constraint ensures correct disentanglement of synthetic images by the shared encoder and the decoder for the synthetic domain

$$\mathcal{L}_{\text{attr}} = \mathbb{E}_{(b_1, b_2, b_3)} \; ||b_3 - C_B(b_1, b_2)||, \quad (b_1, b_2, b_3) \sim \mathcal{X}_B \tag{4.2}$$

(c) the cycle constraint was shown [127] to improve semantic consistency in visual correspondences learned by unsupervised image-to-image translation models

$$\mathcal{L}_{\text{cyc}} = \mathbb{E} \; ||a - C_A(\tilde{b}_c, \tilde{b}_c)|| + \mathbb{E} \; ||b - C_B(\tilde{a}_c, \tilde{a}_c)|| \tag{4.3}$$

$$\tilde{b}_c = C_B(a, a), \quad \tilde{a}_c = C_A(b, b), \quad a \sim \mathcal{X}_A, \quad b \sim \mathcal{X}_B \tag{4.4}$$

(d) the pair of attribute cycle constraints prevents shared encoders and the real decoder G_A from converging to a degenerate solution - decoding the entire real image from a single embedding and completely ignoring the other part. The first "attribute cycle constraint" (the left column in Figure 4·5d) ensures that the first argument of C_A is not discarded:

$$\mathcal{L}^{\text{cyc}}_{\text{attr-B}} = \mathbb{E} \; ||b_3 - C_B(\tilde{a}, b_2)|| \tag{4.5}$$

$$\tilde{a} = C_A(b_1, a), \quad a \sim \mathcal{X}_A, \quad (b_1, b_2, b_3) \sim \mathcal{X}_B. \tag{4.6}$$

The only thing that is important about $\tilde{a}$ as the first argument of C_B is its attribute value, so C_A must not discard the attribute value of its *first* argument b_1, since otherwise, reconstruction of b_3 would become impossible. The "rest" component of a should not influence the estimate of b_3 since it only affects the "rest" component of $\tilde{a}$ that is discarded by later application of C_B. To ensure that the second "rest embedding" argument of C_A is not always discarded, the second attribute cycle constraint (the right column in Figure 4·5d)

$$\mathcal{L}^{\text{cyc}}_{\text{attr-A}} = \mathbb{E}\ ||a - C_A(\tilde{b}, a)|| \tag{4.7}$$

$$\tilde{b} = C_B(a, b), \quad a \sim \mathcal{X}_A, \quad b \sim \mathcal{X}_B \tag{4.8}$$

penalizes C_A if it ignores its *second* argument since the "rest" of a is not recorded in $\tilde{b}$ and therefore can be obtained by C_A only from its second argument.

The proposed method can be easily extended to disentangle multiple attributes at once using separate encoders and example triplets for each attribute. For example, to disentangle two attributes p and q using encoders E^p_{attr}, E^q_{attr} and synthetic triplets $(b^p_1, b^p_2, b^p_3), (b^q_1, b^q_2, b^q_3)$ where (b^p_2, b^p_3) share all other attributes except p (*including* q), and vice versa, the disentanglement constraint should look like:

$$b^p_3 = G_B(E^p_{attr}(b^p_1), E^q_{attr}(b^p_2), E_{rest}(b^p_2)) \tag{4.9}$$

$$b^q_3 = G_B(E^p_{attr}(b^q_2), E^q_{attr}(b^q_1), E_{rest}(b^q_2)). \tag{4.10}$$

In addition to the supervised losses described above, we apply unsupervised adversarial LS-GAN [81] losses to all generated images. Discriminators $D_K(x')$ and attribute combination operators $C_K(x, y)$ are trained in an adversarial fashion so that any combination of embeddings extracted from images x, y from either of two domains and decoded via either real or synthetic decoder G_K looks like a reasonable sample from the respective domain.

Architecture. We used the "CycleGAN resnet" encoder (padded 7x7 conv followed by two 3x3 conv with stride 2 all with relus), followed by six residual conv blocks (two 3x3 convs with relus) a fully-connected bottleneck of size 128 and a pix2pix decoder (two bi-linear up-sampling followed by a convolution). We used LS-GAN objective in all GAN losses. It generally follows the architecture of CycleGAN implementation provided in the tfgan package[1].

Training. We optimized the entire loss jointly with respect to all encoder-decoder weights and then all discriminator losses in two consecutive iterations of the Adam optimizer with $\alpha = $ (2e-4, 5e-5) learning rates with polynomial decay and $\beta = 0.5$. A model trained by updating different losses wrt different weights independently in an alternating fashion did not converge, so all generator and discriminator losses must be updated together in two large steps. We also added Gaussian instance noise to each image used in disentanglement and attribute cycle losses to improve stability during training. We added stop gradient op after the application of C_B in the second attribute cycle loss and instance noise to all intermediate images to avoid the "embedding" behavior. We purposefully avoid constraining embeddings themselves, *e.g.* penalizing Euclidean distances between embedding components of images that are known to share a particular attribute, as such penalties often cause embedding magnitudes to vanish.

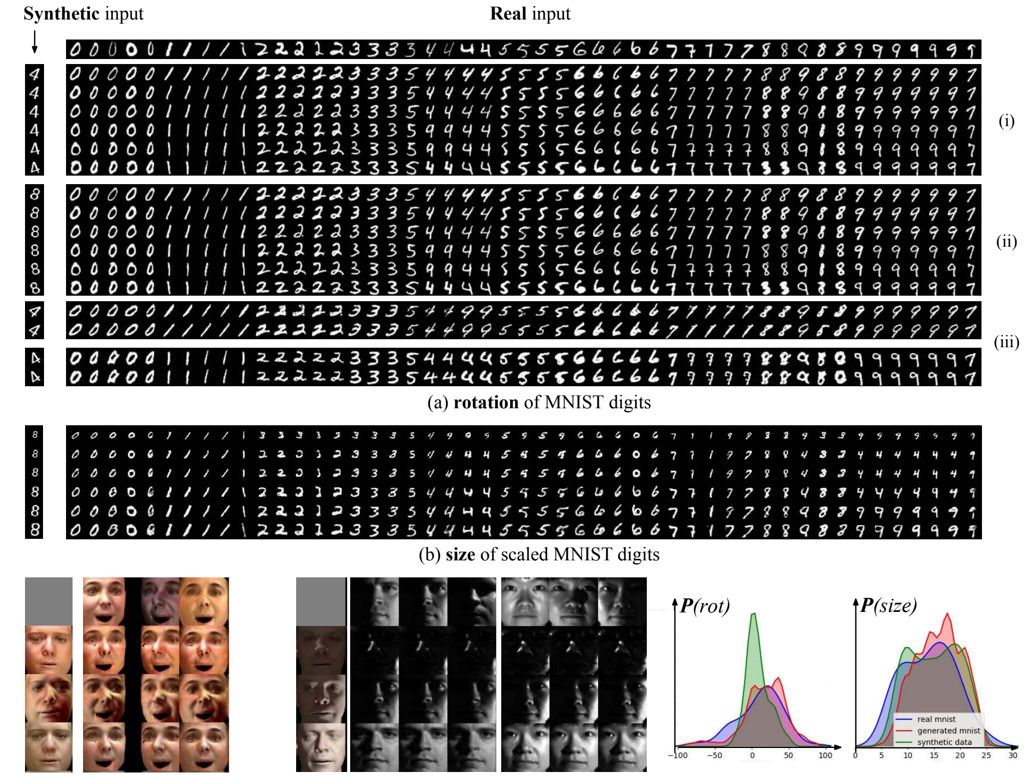
4.1.3 Experiments

Setup. We evaluated the ability of our model to disentangle and manipulate individual attributes of real images using synthetic demonstrations in multiple different settings illustrated in Figures 4·4 and 4·6.

1. *Size and rotation of real digits* from MNIST and USPS were manipulated using a synthetic dataset of typewritten digits rendered using a sans-serif Roboto font.

2. *Mouth expression in human face crops* from the VW-300 [103] dataset was manipulated using synthetic face renders with varying face orientation and expression, but same identity and lighting, obtained using Basel parametric face model [59] with the

[1] https://www.tensorflow.org/api_docs/python/tf/contrib/gan/CycleGANModel



(a) **rotation** of MNIST digits

(b) **size** of scaled MNIST digits

(c) **spherical harmonic lighting**

(d) **light direction and intensity** in YaleB

(e) **distributions** of attribute values

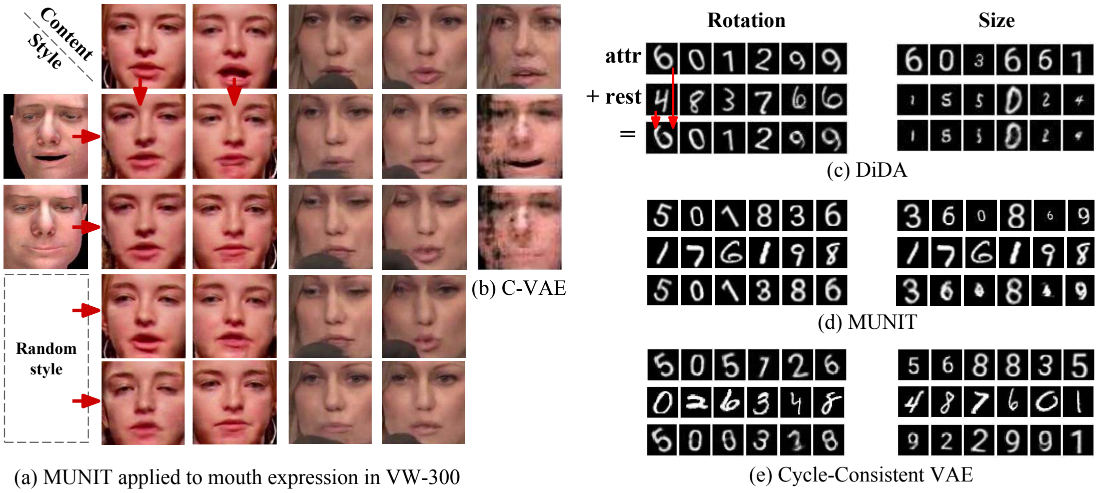
Figure 4·6: **(a-d)** The PuppetGAN model correctly transfers synthetic AoI onto real images and completely ignores other attributes of synthetic inputs. **(ii)** For example, in the digit rotation experiment, when a synthetic input with *the same rotation* but a different size and class label (e.g. smaller "eight" instead of bigger "four") is passed through the model, the outputs *do not change*. **(iii)** Our model is robust to synthetic inputs with AoI (rotation) beyond the range observed during training - it "saturates" on synthetic outliers. **(e)** The distribution of attributes is monotonically remapped to match the real domain.

global illumination prior [27].

3. *Global illumination* (spherical harmonics) in female synthetic face renders was manipulated using male renders with different head orientations and expressions.

4. *Direction and power of the light source* in real faces from the YaleB [33] dataset were manipulated using synthetic 3D face renders with varying lighting and identities (but the constant expression and head orientation).

We used visually similar digit dataset pairs to investigate how discrepancy in attribute distributions affects the performance of the model, *e.g.* how it would perform



(a) MUNIT applied to mouth expression in VW-300

(b) C-VAE

(c) DiDA

(d) MUNIT

(e) Cycle-Consistent VAE

Figure 4·7: Results of methods related to PuppetGAN (only DiDA is directly comparable) **(a)** MUNIT disentangled mouth expression from head orientation, but style spaces of two domains are not aligned, so controlled mouth manipulation is not possible; **(b)** Cycle-Consistent VAE is not suited for large domains shift; **(c)** DiDA converged to degenerate solutions that used only one input; **(d)** MUNIT disentangled stroke from other attributes (i.e. did not isolate rotation or size from the class label); **(e)** Cycle-Consistent VAE was able to extract information only from real inputs that looked "synthetic enough".

if synthetic digits looked similar to real digits, but were much smaller than real ones or rotated differently. In face manipulation experiments we used a much more visually distinct synthetic domain. In VW-300 experiments we treated each identity as a separate domain, so the model had to learn to preserve head orientation and expression of the real input; we used the same set of 3D face renders across all real identities. In the experiment on reapplying environmental lighting to synthetic faces, the expression and head orientation of the input had to be preserved. In the lighting manipulation experiment on the YaleB dataset, we used a single large real domain with face crops of many individuals with different lighting setups each having the same face orientation across the dataset, so the model had to learn to disentangle and preserve the identity of the real input.

Metrics. In order to quantitatively evaluate the performance of our model on digits we evaluated Pearson correlation (r) between measured attribute values in inputs and generated images. We measured the rotation and size of both input and generated digit images using image moments, and trained a LeNet [60] to predict digit class. Below we define the metrics reported in Table 4.2. The AoI measurements in images generated by an "ideal" model should strongly correlate with the AoI measurements in respective synthetic inputs ($r^{\text{syn}}_{\text{attr}}$ ↑ - the arrow direction indicates if larger or smaller values of this metric is "better"), and the measurement of other attributes should strongly correlate with those in real inputs (Acc - accuracy of preserving the digit class label - higher is better), and no other correlations should be present ($r^{\text{syn}}_{\text{rest}}$ lower is better). For example, in digit rotation experiments we would like the rotation of the generated digit to be strongly correlated with the rotation of the synthetic input and uncorrelated with other attributes of the synthetic input (size, class label, etc.); we want the opposite for real inputs. Also, if we use a different synthetic input with the same AoI value (and random non-AoI values) there should be no change in pixel intensities in the generated output (small variance V_{rest}). Optimal values of these metrics are often unachievable in practice since attributes of real images are not independent, *e.g.* inclination of real digits is naturally coupled with their class label (sevens are more inclined than twos), so preserving the class label of the real input inevitably leads to a non-zero correlation between rotation measurements in real and generated images. We also estimated discrepancy in attribute distributions by computing Jensen-Shannon divergence between optimal [106] kernel density estimators of respective attribute measurements between real and synthetic images (J^{syn}) as well as real and generated images (J^{gen}). In order to quantitatively evaluate to what extent proposed disentanglement losses improve the quality of attribute manipulation, we report the same metrics for an analogous model without disentanglement losses that

Model	Disentanglement Quality							
	Size				Rotation			
	Acc ↑	$r^{\text{syn}}_{\text{attr}}$ ↑	$r^{\text{syn}}_{\text{rest}}$ ↓	V_{rest} ↓	Acc ↑	$r^{\text{syn}}_{\text{attr}}$ ↑	$r^{\text{syn}}_{\text{rest}}$ ↓	V_{rest} ↓
PuppetGAN	**0.73**	**0.85**	**0.02**	**0.02**	**0.97**	**0.40**	**0.11**	**0.01**
CycleGAN [127]	0.10	0.28	**0.06**	0.28	0.11	**0.54**	0.37	0.33
DiDA [15]	**0.71**	0.18	0.09	**0.02**	**0.86**	0.04	0.35	**0.02**
MUNIT [47]	**0.96**	0.06	0.09	**0.01**	**1.00**	0.00	0.15	**0.01**
Cycle-VAE [41]	0.17	**0.92**	0.16	**0.01**	0.29	**0.45**	**0.10**	**0.01**
PuppetGAN†	**0.64**	0.28	0.07	**0.01**	0.10	0.06	**0.04**	**0.01**

Table 4.2: Rotation and scaling of MNIST digits (Figures 4·6-4·7). Our model exhibits a higher precision of attribute manipulation. We measure how well models preserve the class labels of real inputs (Acc), AoI of synthetic inputs $r^{\text{synth}}_{\text{attr}}$, and ignore non-AoI of synthetic inputs $r^{\text{synth}}_{\text{rest}}$. We investigate how increased discrepancy between sizes of synthetic and real digits (meaning higher $J^{\text{syn}}_{\text{attr}}$ for size and $J^{\text{syn}}_{\text{rest}}$ for rotation) affects the performance of our model (PuppetGAN†). Arrows ↑↓ indicate if higher or lower values are better, good results are **underscored**.

translates all attributes of the synthetic input to the real domain (CycleGAN).

Hyperparameters. We did not change any hyperparameters across tasks, the model performed well with the initial "reasonable" choice of parameters listed in the supplementary. As with all adversarial methods, our model is sensitive to the choice of generator and discriminator learning rates.

4.1.4 Results.

The proposed model successfully learned to disentangle the attribute of interest (AoI) and enabled isolated manipulation of this attribute using embeddings of synthetic images in all considered experiment settings:

1. In the digit rotation experiment (Figure 4·6a), generated images had the class label, size, and style of the respective real input and rotation of the respective synthetic input, and did not change if either class or size of the synthetic (Figure 4·6(ii)), or rotation of the real input changed. Attributes were properly disentangled in all face manipulation experiments (Figure 4·4ab, 4·6cd), *e.g.* in the YaleB experiment "original" lighting of real faces and identities of synthetic faces did not affect the output,

whereas identities of real faces and lighting of synthetic faces were properly preserved and combined. For the VW-300 domain with face crops partially occluded by a microphone, the proposed model preserved the size and position of the microphone, and properly manipulated images with the partially occluded mouth, even though this attribute was not modeled by the simulation.

2. Larger discrepancy between attribute distributions in two domains (PuppetGAN† in Table 4.2) leads to poorer attribute disentanglement, *e.g.* if synthetic digits are much smaller than real, or much less size variation is present in the real MNIST, or much less rotation in USPS (see the published paper [119]). For moderate discrepancies in attribute distributions, AoI in generated images followed the distribution of AoI in the real domain (Figure 4·6e, and supplementary of the original paper [119]). If during evaluation the property of interest in a synthetic input was beyond values observed during training, the model's outputs "saturated" (Figure 4·6(iii)).

3. Ablation study results (supplementary of the original paper [119]) and the visual inspection of generated images suggest that domain-agnostic encoders help to semantically align embeddings of attributes across domains. Image level GAN losses improve the "interchangeability" of embedding components from different domains. Learned representations are highly excessive, so even basic properties such as "digit rotation" required double-digit embedding sizes. Attribute cycle losses together with pixel-level instance noise in attribute and disentanglement losses improved convergence speed, stability, and the resilience of the model to degenerate solutions [7].

Comparison to Related Methods. To our knowledge, only E-CDRD [73] and DiDA [15] considered similar input constraints at train time (both use explicit labels). We could not obtain any implementation of E-CDRD, and since authors focused on different applications (domain adaptation for digit classification, manipulation of photos using sketches), their reported results are not comparable with ours. While

MUNIT [47] (unsupervised cross-domain) and Cycle-Consistent VAE [41] (single-domain) methods have input constraints incompatible with ours, we investigated how they perform, respectively, without attribute supervision and in the presence of the domain shift. Quantitative evaluation (Table 4.2) supports our explanations of qualitative results (Figures 4·6-4·7). Proposed losses greatly improve the quality of isolated attribute manipulation over both cross-domain non-disentangled (Cycle-GAN), cross-domain disentangled (DiDA, MUNIT), and single-domain disentangled (Cycle-VAE) baselines. More specifically, MUNIT disentangled the wrong attribute (stroke) and DiDA converged to degenerate solutions that ignored synthetic AoI - both have low $r^{\text{syn}}_{\text{attr}}$. The Cycle-VAE disentangled correct attributes of digits (high $r^{\text{syn}}_{\text{attr}}$), but due to the domain shift failed to preserve class labels of real inputs (low Acc). Figure 4·7a shows that MUNIT disentangled face orientation as “content” and mouth expression as “style”, as random style vectors appear to mostly influence the mouth. Unfortunately, style embedding spaces of two domains are not semantically aligned, so controlled manipulation of specific attributes (e.g. mouth) across domains is not possible. The available implementation of DiDA made it very difficult to apply it to faces. Cycle-Consistent VAE learned great disentangled representations and enabled controlled manipulation of *synthetic* images, but, like in digits experiments, failed to encode and generate plausible real faces because domains looked too different (Figure 4·7b).

Conclusion. In this section we presented a novel task of “cross-domain image manipulation by demonstration” and a model that excels in this task on a variety of realistic and proof-of-concept datasets. Our approach enables controlled manipulation of real images using crude simulations, and therefore can immediately benefit practitioners that already have imprecise models of their problem domains by enabling controlled manipulation of real data using existing imprecise models.

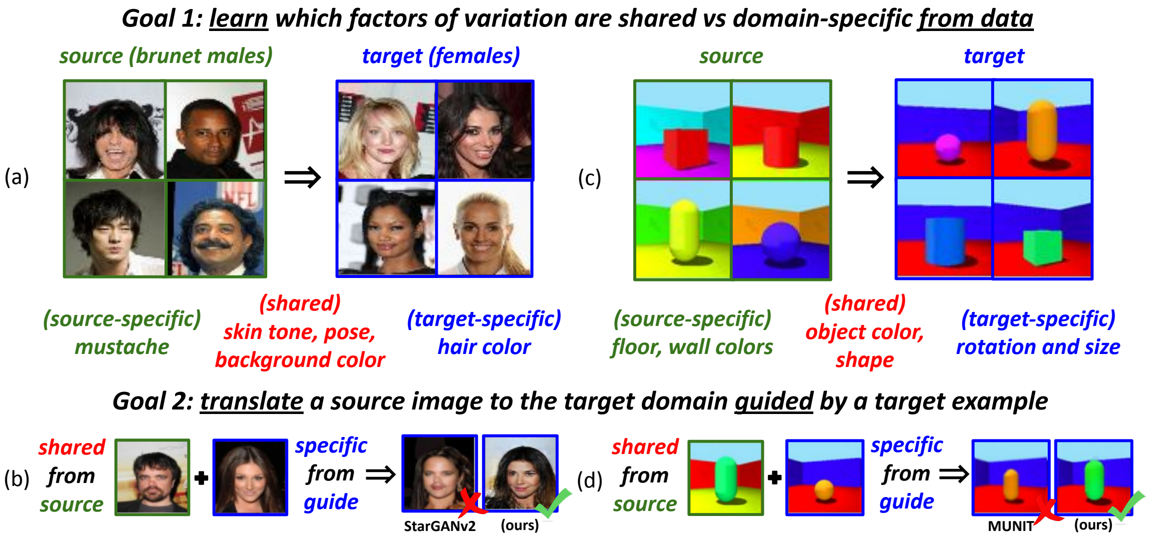
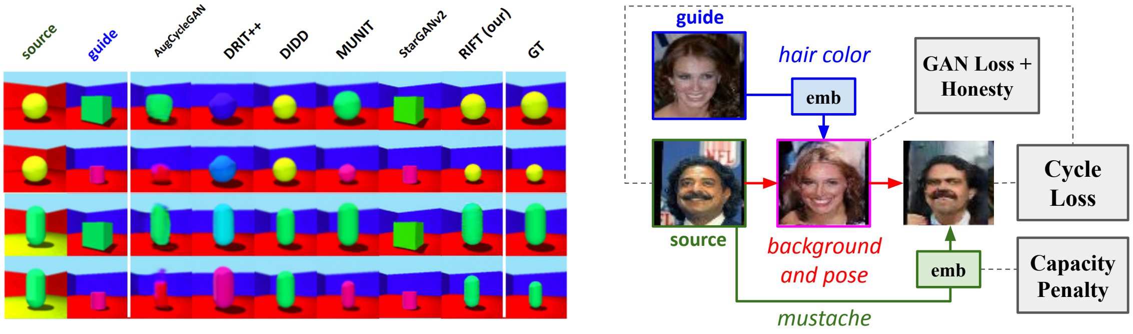

Goal 1: learn which factors of variation are shared vs domain-specific from data
source (brunet males)
target (females)
source
target
(a)
(c)
(source-specific) mustache
(shared) skin tone, pose, background color
(target-specific) hair color
(source-specific) floor, wall colors
(shared) object color, shape
(target-specific) rotation and size
Goal 2: translate a source image to the target domain guided by a target example
(b)
shared from source
specific from guide
StarGANv2
(ours)
(d)
shared from source
specific from guide
MUNIT
(ours)

Figure 4·8: An unsupervised many-to-many image translation model must *disentangle* factors of variation shared across two domains from those specific to each domain using *unpaired* sets of source and target images during training. At the same time, the model has to perform domain translation, preserving factors of the source image shared across two domains and applying target-specific factors from the "guide" image. We show that existing methods fail on at least one of the two datasets shown above, and the proposed method excels on both.

4.2 Disentangling Domain-Specific and Shared Factors

As illustrated in Figure 4·1c, we can not use simulation to manipulate factors of real images absent from that simulation, but in this section, we show that we still can learn something about these factors from unpaired image sets. More specifically, we show how to train a model that can differentiate domain-specific factors of variation from those shared across two domains and manipulate these two groups of factors in independently from each other.

When one domain has *unique factors of variation* absent in the other domain, in order to keep the task well-defined, we must alter the definition of the unsupervised alignment problem from mapping individual source images to the target domain - to mapping *pairs* of source and target images to the target domain. The resulting



(a) Flaws in existing methods.

(b) An overview of RIFT.

Figure 4·9: (a) On Shapes-3D-A, shown in Fig. 4·8(c-d), all prior methods fail to either preserve shared attributes of the source (shape, object color), or apply target-specific attributes of the guide (size, orientation), while the proposed method (RIFT) succeeds at both (compare to GT). **(b)** To minimize the cycle-reconstruction loss, RIFT encodes source-specific factors of variation (mustache) into the source-specific **embedding**, because, unlike shared factors (background, pose), source-specific factors can not be inferred from an image translated into the target (female) domain, and vice versa.

unsupervised *many-to-many* translation problem [47] has a unique and well-defined solution. More specifically, for an input pair consisting of a source image and a target "guide" image, the learned mapping must generate a new image from the target domain, preserving all factors of variation of the source image that are shared across two domains, and taking factors of variation specific to the target domain from the guide image. For example, in Fig. 4·8(a-b) the task is to preserve the pose, skin tone, and background of the male source, and apply the hair color of the female guide, whereas in Fig. 4·8(c-d), object color and shape should be preserved, and the orientation and size should come from the guide.

Identifying and preserving shared factors during translation is of crucial importance in many applications of unsupervised many-to-many translation, such as preserving skin color for clothing or makeup try-on [65] or face manipulation with synthetic data [119]. In simulation-to-real adaptation [52], it is important to identify the factors present in the real data, but not reflected in the simulation.

Unfortunately, in Section 4.2.2, we show that all state-of-the-art methods fail to infer which attributes are domain-specific and which are domain-invariant *from data* on certain kinds of attribute combinations, and rely on heuristics that work for some dataset pairs, but fail on others. More specifically, many state-of-the-art methods [20, 47] implicitly assume that all domain-specific variations can be modeled as "style vectors" mixed-in globally into intermediate features of image decoders via adaptive instance normalization (AdaIN) [46] originally designed for style transfer. As a result, these methods change *all* colors and textures of the source input to match the guide, even if these colors and textures are varied across both domains and therefore should be preserved. For example, Fig. 4·9a shows that even on a toy dataset pair from Fig. 4·8(c-d), MUNIT and StarGANv2 change the color of the source object to match the color of the guide, even though the object color should be preserved. In Sec. 4.2.6, we show that these methods also change backgrounds and skin tones in the female-to-male setup from Fig. 4·8(a-b), even though they must be preserved. On the other hand, methods based on auto-encoders and reconstruction losses [1, 11, 63] preserve shared information better, but often fail to apply correct domain-specific factors. For example, in Fig. 4·9a DIDD [11] preserved the object color of the source, but failed to extract and apply the correct orientation and size from the guide.

4.2.1 Background

Many-to-many translation. To account for (and enable control over) domain-specific factors, many-to-many image translation methods [1, 20, 47, 63, 72] separate domain-invariant "content" from domain-specific "style". We avoid the terms "content" and "style" to distinguish the general many-to-many translation problem from its subtask - style transfer [32].

Adaptive instance normalization. Many state-of-art many-to-many translation methods [20, 47], use AdaIN [46], originally designed for style transfer. Some methods [80] add a spatial dimension to AdaIN to distinguish colors and textures of different objects, but fundamentally still rely on the re-normalization of decoder features to perform disentanglement. While effective at realistic layout-preserving texture transfer (day-to-night, summer-to-winter), this architectural choice was shown [8] to limit the range of applications of these methods to cases when domain-specific information lies within textures and colors.

Autoencoders. In contrast, methods like Augmented CycleGAN [1], DRIT++ [63], and Domain Intersection and Domain Difference (DIDD) [11] rely on embedding losses and therefore are more general. For example, DIDD forces domain-specific embeddings of the opposite domain to be zero, while DRIT++ uses adversarial training to make the source and target content embeddings indistinguishable.

Cycle losses. Most many-to-many methods [1, 47] use cycle-consistency on domain-specific embeddings to ensure that information about the guide image is not ignored during translation, and cycle loss on reconstructed images [127] to improve semantic consistency. However, cycle losses on images have been shown [7, 21] to force one-to-one unsupervised translation models to "cheat" by hiding domain-specific attributes into translations in the form of imperceptible low-amplitude structured noise. Alternative consistency objectives, such as the patchwise contrastive loss [94], are designed to be invariant to differences across domains, and therefore can not be used to supervise the manipulation of domain-specific factors in the many-to-many case.

Few-shot [72] and **truly unsupervised** [5] translation methods solve a related but different problem. Since these methods have either very few domain examples or no domain labels whatsoever, shared and domain-specific attributes can not be inferred (or even defined) by looking at data. To resolve this ambiguity, these methods also assume that the layout distribution is shared and that the variability in appearance (*e.g.* colors and textures) is domain-specific.

Single-domain unsupervised disentanglement methods, such as InfoGAN [18] and β-VAE [43], tackle a different problem as well. First, many-to-many translation is not aimed at in controlled manipulation of *individual factors*, but of all domain-specific or all shared factors at once. Second, if we applied these methods blindly to the combined source and target dataset to analyze the distribution of latent codes across each domain, the structure of the combined dataset would conflict with the core assumption of independence of latent features built into these methods, since distributions of domain-specific factors are *not independent* both from each other and from the distribution of domain labels.

Overall, prior methods ensure that the guide input modulates the translation result in some non-trivial way, but, to our knowledge, no prior work explicitly addresses the adversarial embedding of domain-specific information into the translated image, or quantitatively verifies that domain-specific factors are correctly applied and shared factors are preserved during translation, and our work discussed in the remainder of this section fills this gap.

4.2.2 Evaluation Protocol

Author Contribution. The evaluation protocol and findings concerning existing many-to-many translation methods described in this subsection were first reported by Bashkirova et al. [8]. Ben Usman helped constructing synthetic datasets, automating evaluation, and writing this paper, but the idea and core technical contribution of the proposed evaluation pipeline paper should be attributed to its first author.

We propose a new, data-driven approach for the evaluation of unsupervised cross-domain disentanglement quality in unsupervised many-to-many image-to-image (UMMI2I) methods. We designed three evaluation protocols based on the synthetic 3D Shapes [55] dataset (originally designed for evaluation of single-domain disentan-

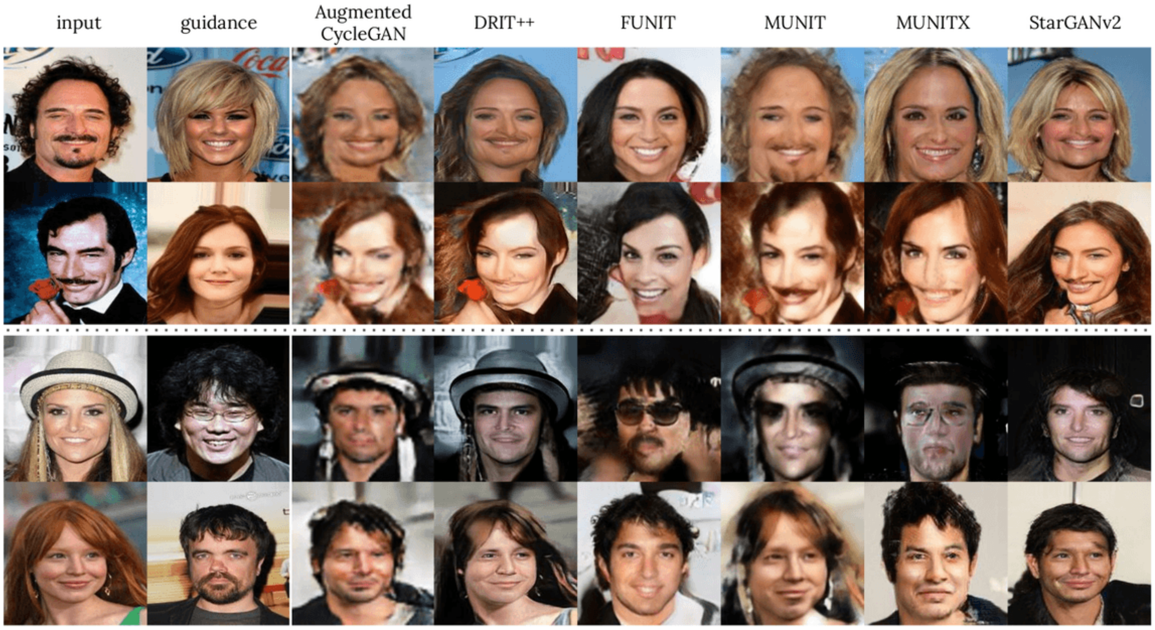


Figure 4·10: Examples of M→F (top) and F→M (bottom) translations on the proposed **CelebA-D** subset. A correct translation should have domain-specific attributes of the guidance image (hair color in the top two lines; facial hair, smile, and age in the bottom two lines), and the rest of the attributes (facial features, orientation, etc.) from the input image.

glement), a more challenging synthetic SynAction [110] pose dataset, and a widely used CelebA [75] dataset of faces.

1. To the best of our knowledge, we are the first to propose a set of metrics for evaluation of the semantic correctness of UMMI2I translation. Our metrics evaluate how well the shared attributes are preserved, how reliably the domain-specific attributes are manipulated, whether the translation result is a valid example of the target domain and whether the network collapsed to producing the same most frequent attribute values.

2. We create three evaluation protocols based on 3D-Shapes, SynAction and CelebA datasets, and measure the disentanglement quality of the current state-of-the-art UMMI2I translation methods on them.

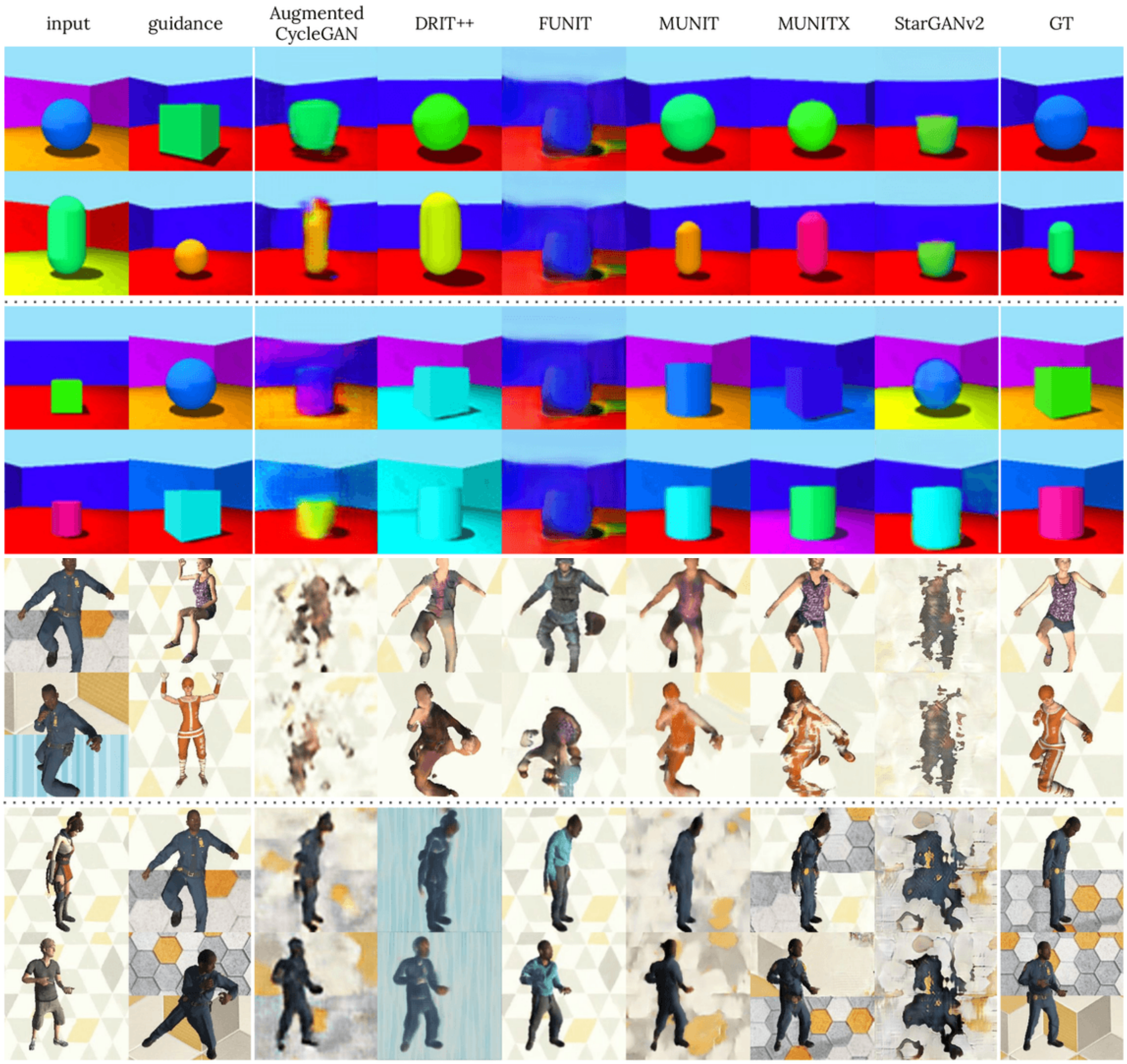


Figure 4·11: Illustration of UMMI2I translation results on **3D-Shapes** (top) and **SynAction** (bottom) subsets. Domain-specific attributes in **3D-Shapes** are wall and floor color (A) / size and view angle (B), and in **SynAction** are background texture (A) / clothing and identity (B). The ground truth (GT) outputs can be found in the rightmost column.

3. We show that for all tested methods there is a clear trade-off between content preservation and manipulation of the domain-specific variations, leading to a subpar performance on at least one dataset. More specifically, all methods we tested poorly manipulated attributes associated with adding or changing certain parts of the objects (e.g. facial hair or smile) and AdaIN-based [46] methods showed an inductive bias toward treating spatial attributes, such as poses and position of objects in the scene, as the domain-invariant factors, and colors and textures as the domain-specific sources of variation, irrespective of which attributes were actually shared between domains and which are domain-specific.

Qualitative results can be found in Figures 4·10 and 4·11 and we refer the reader to the original paper [8] for qualitative evaluation.

4.2.3 Restricted Information Flow for Translation

Setup. Following Huang et al. [47], we assume that we have access to two unpaired image datasets $A = \{a_i\}$ and $B = \{b_i\}$ that share some semantic structure, but differ visually (*e.g.* male and female faces with poses, backgrounds and skin color varied in both). In addition to that, each domain has some attributes that vary only within that domain, *e.g.* only males have variation in the amount of facial hair and only females have variation in the hair color (Fig. 4·8). Our goal is to find a pair of guided cross-domain mappings $F_{\text{A2B}} : A, B \to B$ and $F_{\text{B2A}} : B, A \to A$ such that for any source inputs a_s, b_s and guide inputs a_g, b_g from respective domains, resulting guided cross-domain translations $b' = F_{\text{A2B}}(a_s, b_g)$ and $a' = F_{\text{B2A}}(b_s, a_g)$ look like plausible examples of respective output domains, share domain-invariant factors with their "source" arguments (a_s and b_s respectively) and domain-specific attributes with their "guidance" arguments (b_g and a_g respectively). For example, the correct guided female-to-male mapping F_{B2A} applied to female source image b_s and a guide male

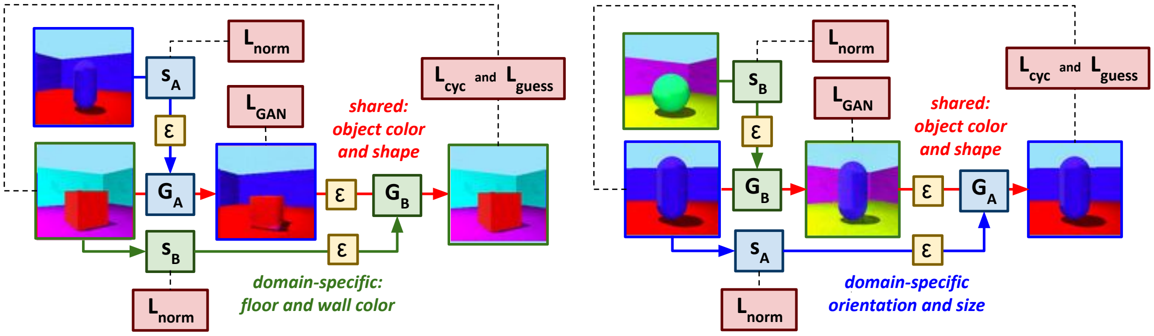

L_norm
S_A
L_GAN
shared: object color and shape
L_cyc and L_guess
ε
G_A
ε
G_B
S_B
ε
domain-specific: floor and wall color
L_norm
L_norm
S_B
L_GAN
shared: object color and shape
L_cyc and L_guess
ε
G_B
ε
G_A
S_A
ε
domain-specific orientation and size
L_norm

Figure 4·12: Losses used to train RIFT. For illustration, we use 3D-Shapes-A described in Section 4.2.5 and illustrated in Fig. 4·13 and Fig.4·9a. When the model is trained, green arrows carry only B-specific information (floor and wall color), blue arrows carry only A-specific information (orientation and size), and red arrows carry information shared across two domains (object color and shape).

image a_g should generate a new male image a' with pose, background, and skin tone from the female input image b_s, and facial hair from the guidance input a_g, because poses, backgrounds and skin tone vary in both, while facial hair is male-specific.

Method. While it might be possible to approximate functions F_{A2B} and F_{B2A} directly, following prior work, we split each one into two learnable parts: encoders $s_A(a), s_B(b)$ that extract domain-specific information from corresponding guide images, and generators $G_{\text{A2B}}(a, s_b)$ and $G_{\text{B2A}}(b, s_a)$ that combine that domain-specific information with a corresponding source image, as illusrated in Figure 4·12. Final many-to-many mappings are just compositions of encoders and generators:

$$F_{\text{A2B}}(a, b) = G_{\text{A2B}}(a, s_B(b)),\ F_{\text{B2A}}(b, a) = G_{\text{B2A}}(b, s_A(a)) \tag{4.11}$$

Components introduced in the remainder of this section ensure that encoders s_* extract all domain-specific information from their inputs and nothing else, and that generators G_* use that information, along with domain-invariant factors from their source inputs to form plausible images from corresponding domains.

Noisy cycle-consistency loss. First, to ensure that each factor of input images is not ignored completely, we use a guided analog of the cycle consistency loss [127]. This loss ensures that any image translated into a different domain, and translated back with its original domain-specific embedding is reconstructed perfectly. Additionally, before translating images back into their original domains, we add zero-mean Gaussian noise (ε_s, ε_g) of amplitudes σ_s and σ_g and appropriate shapes to generated images and domain-specific embeddings respectively - the motivation is given in the two following paragraphs.

$$L^A_{\text{cyc}} = \mathbb{E}_{a,b}\ ||a_{\text{cyc}} - a||_1, \quad L^B_{\text{cyc}} = \mathbb{E}_{b,a}\ ||b_{\text{cyc}} - b||_1 \tag{4.12}$$

$$a_{\text{cyc}} = G_{\text{B2A}}(G_{\text{A2B}}(a, s_B(b) + \varepsilon_g) + \varepsilon_s, s_A(a) + \varepsilon_g) \tag{4.13}$$

$$b_{\text{cyc}} = G_{\text{A2B}}(G_{\text{B2A}}(b, s_A(a) + \varepsilon_g) + \varepsilon_s, s_B(b) + \varepsilon_g) \tag{4.14}$$

$$a \sim A,\ b \sim B,\ \varepsilon_s \sim \mathcal{N}(0, \sigma_s),\ \varepsilon_g \sim \mathcal{N}(0, \sigma_g) \tag{4.15}$$

Translation honesty. Unfortunately, any form of cycle loss encourages the model to “hide” domain-specific information inside the translated image in the form of structured adversarial noise [21]. To prevent the model from “hiding” the domain-specific information, such as mustache, inside a generated female image (instead of putting it into a male-specific embedding s_a), we use two so-called “self-adversarial defenses” proposed by Bashkirova et al. [7]. First, we *destroy* the structured signal by adding Gaussian noise ε_s to intermediate images before cycle reconstruction, see Eq. (4.13) above. Moreover, we use an additional *guess loss* to train the generator. To compute it, we train a pair of *guess discriminators* that predict which of its two inputs is a cycle-reconstruction and which is the original image. For example, if the male-to-female generator G_{A2B} is consistently adversarially embedding mustaches into all generated female images, then the cycle-reconstructed female b_{cyc} will also have traces of an embedded mustache, because it was generated using that male-to-female generator G_{A2B}, and will be otherwise identical to the input b. In this case, the guess discriminator D_B^{gs}, trained specifically to detect differences between input images and their cycle-reconstructions, will detect this hidden signal and penalize the model:

$$L_{\text{guess}}^A = [D_A^{\text{gs}}(a, a_{\text{cyc}})]^2 + [1 - D_A^{\text{gs}}(a_{\text{cyc}}, a)]^2 \tag{4.16}$$

$$L_{\text{guess}}^B = [D_B^{\text{gs}}(b, b_{\text{cyc}})]^2 + [1 - D_B^{\text{gs}}(b_{\text{cyc}}, b)]^2 \tag{4.17}$$

Domain-specific channel capacity. Unfortunately, neither of the two losses described above can prevent the model from learning to embed the entire guide image a_g into the domain-specific embeddings s_a and reconstructing it from that embedding in G_{B2A}, ignoring its first argument completely, *i.e.* just always producing the guide input exactly. In order to prevent this from happening, we add Gaussian noise ε_g to predicted domain-specific embeddings before cycle reconstruction (see Eq. 4.13 above) and penalize norms of these embeddings:

$$L_{\text{norm}}^A = \mathbb{E}_a \; ||s_A(a)||_2^2, \quad L_{\text{norm}}^B = \mathbb{E}_b \; ||s_B(b)||_2^2 \tag{4.18}$$

As we show below, this constrains the *effective capacity* of domain-specific embeddings. Intuitively, the mutual information between the input guide image a_g and the predicted translation a' corresponds to the maximal amount of information that an observer could learn about translations a' by observing guides a_g if they had infinite amount of examples to learn from. Formally, we can show that if we add Gaussian noise of amplitude σ_g and penalize the norms of these embeddings as described above, this mutual information is bounded by:

$$\text{MI}(a_g; a') \lesssim \dim(s_A(a)) \cdot \log_2\left(1 + L_{\text{norm}}^A / \sigma_g^2\right), \tag{4.19}$$

$$\text{where } a' = G_{\text{B2A}}(b_s, s_A(a_g) + \varepsilon_g), \;\; \varepsilon_g \sim \mathcal{N}(0, \sigma_g) \tag{4.20}$$

(see proof in the next Sec. 4.2.4) meaning that minimizing L^{A}_{norm} loss effectively limits the amount of information from the guide image a_g that G_{A2B} can access to generate a', *i.e.* the effective capacity of the domain-specific embedding. Note that disabling either the noise ($\sigma_g = 0$) or the capacity loss ($L_{\text{norm}} \to \infty$) results in effectively *infinite* capacity, so we need both. Intuitively, this bound describes the expected number of "reliably distinguishable" embeddings that we can pack into a ball of radius $\sqrt{L^{A}_{\text{norm}}}$ assuming that each embedding is perturbed randomly by Gaussian noise with amplitude σ_g.

Realism losses. Remaining losses are analogous to the original GAN and identity losses [71] ensuring that generated images lie within respective domains:

$$L^{A}_{\text{GAN}} = [D_A(a)]^2 + [1 - D_A(G_{\text{B2A}}(b, s_A(a) + \varepsilon^{a}_{s}))]^2 \tag{4.21}$$

$$L^{B}_{\text{GAN}} = [D_B(b)]^2 + \big[1 - D_B(G_{\text{A2B}}(a, s_B(b) + \varepsilon^{b}_{s}))\big]^2 \tag{4.22}$$

$$L^{A}_{\text{idt}} = \mathbb{E}_a \; ||G_{\text{B2A}}(a, s_A(a) + \varepsilon_g) - a||_1, \tag{4.23}$$

$$L^{B}_{\text{idt}} = \mathbb{E}_b \; ||G_{\text{A2B}}(b, s_B(b) + \varepsilon_g) - b||_1 \tag{4.24}$$

Discriminator losses We also train discriminators D_A, D_B and guess discriminators $D^{\text{gs}}_{A}, D^{\text{gs}}_{B}$ by minimizing corresponding adversarial LS-GAN [81] losses.

4.2.4 Derivation of the capacity

Let A and B be arbitrary datasets, s and G be domain-specific embedding and generator functions, and a' be the translation from source b to domain A, guided by the target example a. The following theorem bounds the amount of information about a that G can access to generate a'.

Theorem 4.2.1. *The effective capacity of the guided embedding, i.e. the capacity of* *the $a \to a'$ channel, i.e. the mutual information* $\mathrm{MI}(a;a')$ *is bounded by:*

$$\mathrm{MI}(a;a') \lesssim \dim(s(a)) \cdot \log_2\left(1 + L/\sigma^2\right), \tag{4.25}$$

$$\textit{where } a' = G(b, s(a)+\varepsilon),\ \varepsilon \sim \mathcal{N}(0,\sigma^2), \tag{4.26}$$

$$\textit{and } L = \mathbb{E}\|s(a)\|_2^2,\ a \sim A,\ b \sim B \tag{4.27}$$

Proof. Applying the data processing inequality

$$X \to Y \to Z \quad \Rightarrow \quad \mathrm{MI}(X;Z) \le \mathrm{MI}(X;Y) \ \wedge\ \mathrm{MI}(X;Z) \le \mathrm{MI}(Y;Z) \tag{4.28}$$

twice to following Markov chains

$$a \to (s(a)+\varepsilon) \to a', \quad a \to s(a) \to (s(a)+\varepsilon) \tag{4.29}$$

gives us

$$\mathrm{MI}(a;a') \le \mathrm{MI}(a;s(a)+\varepsilon) \le \mathrm{MI}(s(a);s(a)+\varepsilon) \tag{4.30}$$

intuitively meaning that the overall pipeline always loses at least as much information as each of its steps. Then expanding the mutual information in terms of the differential entropy $h(X)$ gives us

$$\mathrm{MI}(s(a);s(a)+\varepsilon) = h(s(a)+\varepsilon) - h(s(a)+\varepsilon|s(a)) \tag{4.31}$$

$$= h(s(a)+\varepsilon) - h(\varepsilon) \tag{4.32}$$

Since the the second raw moment (aka power) of $s(a)$ is bounded by L, the entropy $h(s(a)+\varepsilon)$ will be maximized if $s(a)$ is a k-dimensional spherical multivariate normal with variance L, where $k = \dim(s(a))$ therefore

$$\mathrm{MI}(s(a);s(a)+\varepsilon) \le h(\mathcal{N}_k(0;L+\sigma^2)) + h(\mathcal{N}_k(0;\sigma^2)) \tag{4.33}$$

$$= \frac{1}{2}\ln\left(\frac{(L+\sigma^2)^k}{\sigma^{2k}}\right) \le k \cdot \log_2\left(1 + L/\sigma^2\right). \tag{4.34}$$

□

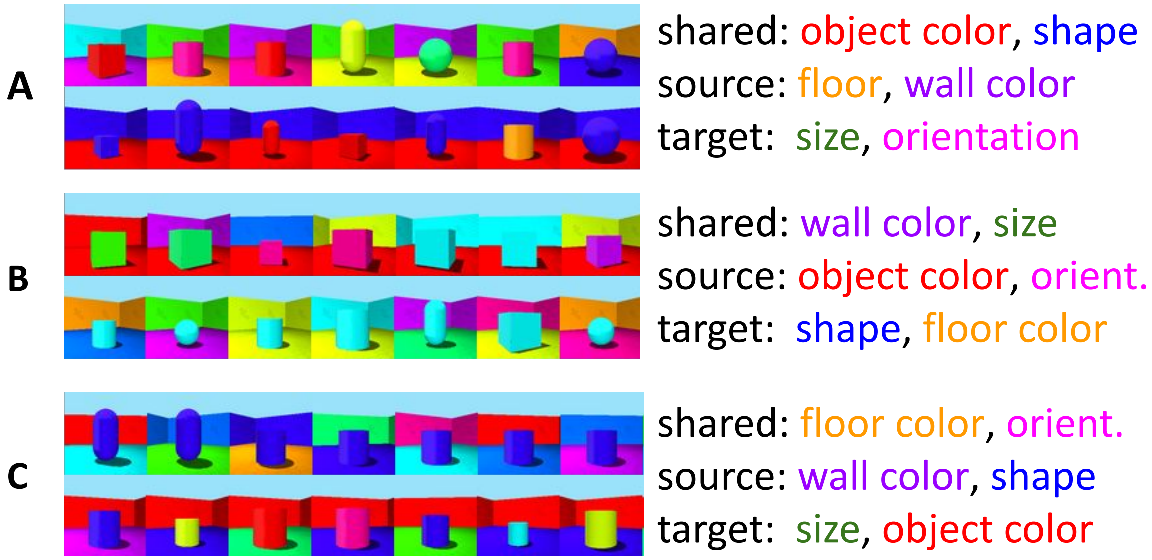

A
shared: object color, shape
source: floor, wall color
target: size, orientation
B
shared: wall color, size
source: object color, orient.
target: shape, floor color
C
shared: floor color, orient.
source: wall color, shape
target: size, object color

Figure 4·13: Shapes-3D-ABC splits. Respective shared and domain-specific attributes.

4.2.5 Experiments

We would like to measure how well each model can generalize across a diverse set of shared and domain-specific attributes. In this section, we discuss datasets we used and generated to achieve this goal, as well as baselines and metrics we used to compare our method to prior work.

Data. Popular image translation datasets (*e.g.* summer-to-winter [71], GTA5-to-BDD, AFHQ [53]) lack attribute annotations, precluding quantitative evaluation, and focus exclusively on layout-preserving texture/palette transfer. To evaluate methods' ability to disentangle and transfer other attributes, following the protocol proposed in Section 4.2.2, we re-purposed existing disentanglement datasets to evaluate the ability of our method to model different attributes as shared and domain-specific. We used 3D-Shapes [55], SynAction, [110] and CelebA [61]. Unfortunately, among the three, only 3D-Shapes [55] is balanced enough and contains enough labeled attributes to make it possible to generate and evaluate all methods across several attribute splits of comparable sizes. For example, if we attempted to build a split of SynAction with a domain-specific pose attribute, the domain with a fixed pose would only contain 90 unique images, which is not sufficient to train an unsupervised translation network.

3D-Shapes-ABC. The original 3D-Shapes [55] dataset contains 40k synthetic images labeled with six attributes: floor, wall, and object colors, object shape and object size, and orientation (viewpoint). There are ten possible values for each color attribute, four possible values for the shape (cylinder, capsule, box, sphere), fifteen values for orientation, and eight values for size. We used three subsets of 3D-Shapes with different attribute splits visualized in Figure 4·13. Three resulting domain pairs contained 4.8k/4k, 12k/3.2k, and 12k/6k images respectively.

SynAction. We used the same split of SynAction [110] as in Section 4.2.2 - with background varied in one domain (nine possible values), identity/clothing varied in the other (ten possible values), and pose varied in both (real-valued vector). The resulting dataset contains 5k images in one domain and 4.6k images in the other. We note that the attribute split of this dataset **matches** the inductive bias of AdaIN methods, since the layout (pose) is shared and textures (background, clothing) are domain-specific in both domains. We noticed that the original “fixed bg” domain introduced in Section 4.2.2 actually has some variations in the background, and fixed them before training both our method and all baselines (see supplementary of the original paper [121]).

CelebA-FM. We used the male-vs-female split discussed in Section 4.2.2 with 25k images in each domain and evaluated the disentanglement of the six most visually prominent attributes: pose, skin, and background color (shared attributes, real-valued vectors), male-specific presence of facial hair (binary), female-specific hair color (three possible values), and domain-defining gender.

Baselines. We compare the proposed method against several state-of-art AdaIN methods, namely MUNIT [47], StarGANv2 [20], MUNITX (Section 4.2.2), and autoencoder-based methods, namely Domain Intersection and Domain Difference (DIDD) [11], Augmented CycleGAN [1], and DRIT++ [63]. We did not evaluate other AdaIN-based methods, such as EGSC-IT [80], since these methods perform disentanglement similarly. We did not evaluate truly unsupervised methods [5] and other methods built *explicitly* to preserve the layout and transfer the appearance [124] because they approach a *different problem*, as discussed in Sec. 4.2.2. We also provide a random baseline (RAND) that corresponds to returning a random image from the target domain to give a sense of scale to reported values.

Metrics. In order to evaluate the performance of our method, we measured how well the domain-specific attributes were manipulated and domain-invariant attributes were preserved. Similar to Section 4.2.2, we trained an attribute classifier $f(x)$, and for each attribute k, we measured its *manipulation accuracy* - the probability of correctly modifying an attribute across input-guide pairs for which the value of the attribute *must change*:

$$\mathrm{ACC}_k^{\mathrm{A}} = p(f_k(F_{\mathrm{A2B}}(a,b)) = y_k^* \mid f_k(a) \neq f_k(b)) \tag{4.35}$$

where the "correct" attribute value equals $y_k^* = f_k(a)$ for shared attributes, and $y_k^* = f_k(b)$ otherwise. For real-valued multi-variate attributes (pose keypoints, background RGB, skin RGB, etc.) we measured the probability of generating an image with predicted attribute vector closer to the correct attribute vector y_k^* then to the incorrect vector y_k' from the opposite domain:

$$\mathrm{ACC}_k^{\mathrm{A}} = p(\|f_k(F_{\mathrm{A2B}}(a,b)) - y_k^*\| \leq \|f_k(F_{\mathrm{A2B}}(a,b)) - y_k'\|) \tag{4.36}$$

where $y_k^* = f_k(a)$ and $y_k' = f_k(b)$ for shared attributes, and vice-versa otherwise. The manipulation accuracy in the opposite direction $\mathrm{ACC}_k^{\mathrm{B}}$ was estimated analogously.

For *Shapes-3D*, we additionally *aggregated* results across three splits by averaging accuracies across splits in which the given attribute was shared/common (C) or domain-specific (S). If we introduce the set of all splits $\mathcal{S}$ and predicates $\text{common}(\text{k}, \text{s})$ and $\text{specific}(\text{k}, \text{s}, \text{dom})$, and the manipulation accuracy at a given split $\text{ACC}_k^A(s)$, *aggregated manipulation accuracy* can be defined as follows:

$$\text{ACC}_k^S = \frac{\sum_{d \in \{\text{A, B}\}} \sum_{s \in S} \text{ACC}_k^d(s) \cdot \text{specific}(\text{k}, \text{s}, \text{d})}{\sum_{d \in \{\text{A, B}\}} \sum_{s \in S} \text{specific}(\text{k}, \text{s}, \text{d})} \tag{4.37}$$

$$\text{ACC}_k^C = \frac{\sum_{d \in \{\text{A, B}\}} \sum_{s \in S} \text{ACC}_k^d(s) \cdot \text{common}(\text{k}, \text{s})}{\sum_{d \in \{\text{A, B}\}} \sum_{s \in S} \text{common}(\text{k}, \text{s})} \tag{4.38}$$

For three splits of *3D-Shapes* we also report the *relative discrepancy* between domain-specific and domain-invariant manipulation accuracies:

$$\text{RD} = 100 \cdot \frac{\sum_k |\text{ACC}_k^S - \text{ACC}_k^C|}{\sum_k (\text{ACC}_k^S + \text{ACC}_k^C)}. \tag{4.39}$$

Evaluation protocol. To compute the metrics above, we generated two guided translations per source image per domain per baseline. We re-ran each method multiple times to account for poor initialization. We used PoseNet [93] to get ground truth poses for SynAction, and Ruiz et al. [101] and median background and skin color for CelebA, see supplementary of the original paper [121] for prediction visualizations.

Architecture. We used standard CycleGAN [21] components: pix2pix [49] generators and patch discriminators with LS-GAN loss [81].

4.2.6 Results

In this section, we first compare our method to prior work both qualitatively and quantitatively. Then we show what happens if we remove key losses discussed in the previous section. And finally, we discuss implicit assumptions made by our method, and key challenges that future methods might encounter in further improving manip-

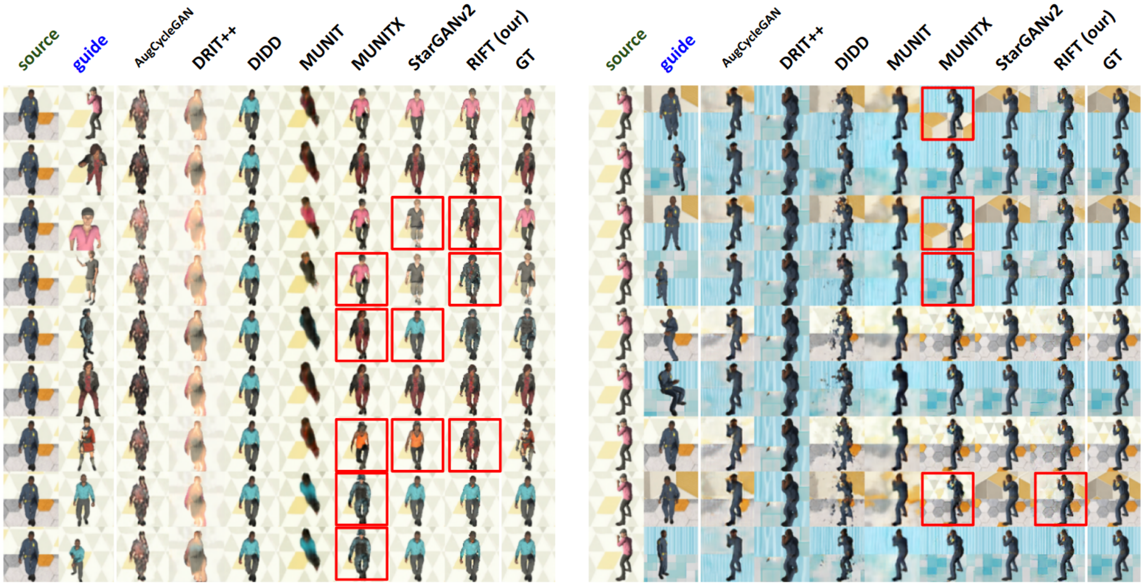


Figure 4·14: Qualitative comparison to prior work on SynAction. Our model correctly preserves shared attributes (pose) of the source image and applies domain-specific attributes of the guide domain (clothing/identity colors on the left, background texture on the right) - compare to Ground Truth (GT). Errors made by top-performing methods are highlighted in red.

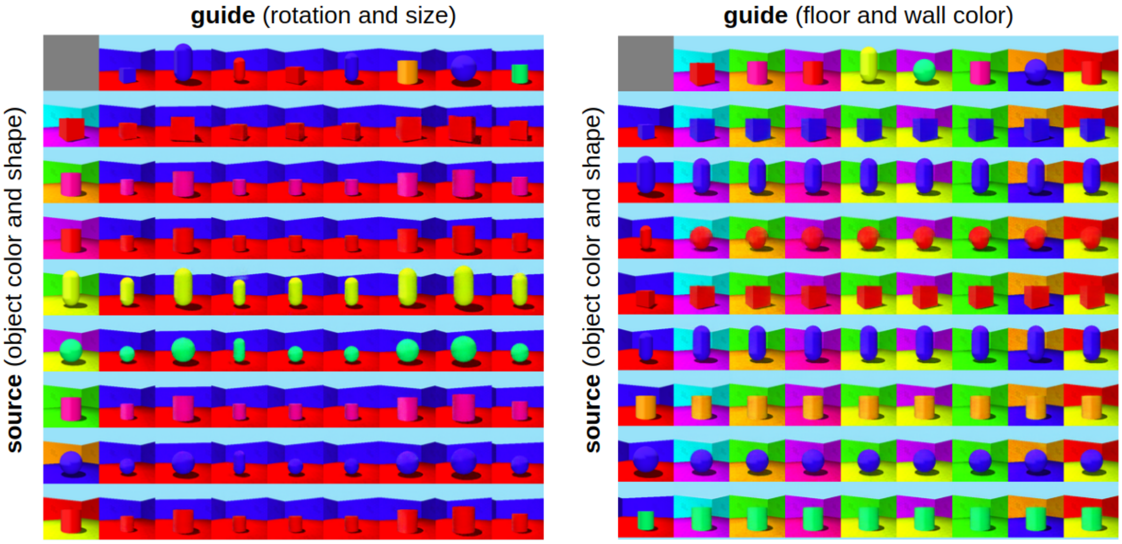

guide (rotation and size)
source (object color and shape)
guide (floor and wall color)
source (object color and shape)

Figure 4·15: Guided translations generated by RIFT on 3D-Shapes-A. Our model successfully preserves shared attributes (object color and shape) of the source image and applies domain-specific attributes of the guide domain (rotation and size on the left, floor and wall color on the right). A qualitative comparison to prior work can be found in Fig. 4·9a and in supplementary of the original paper [121].

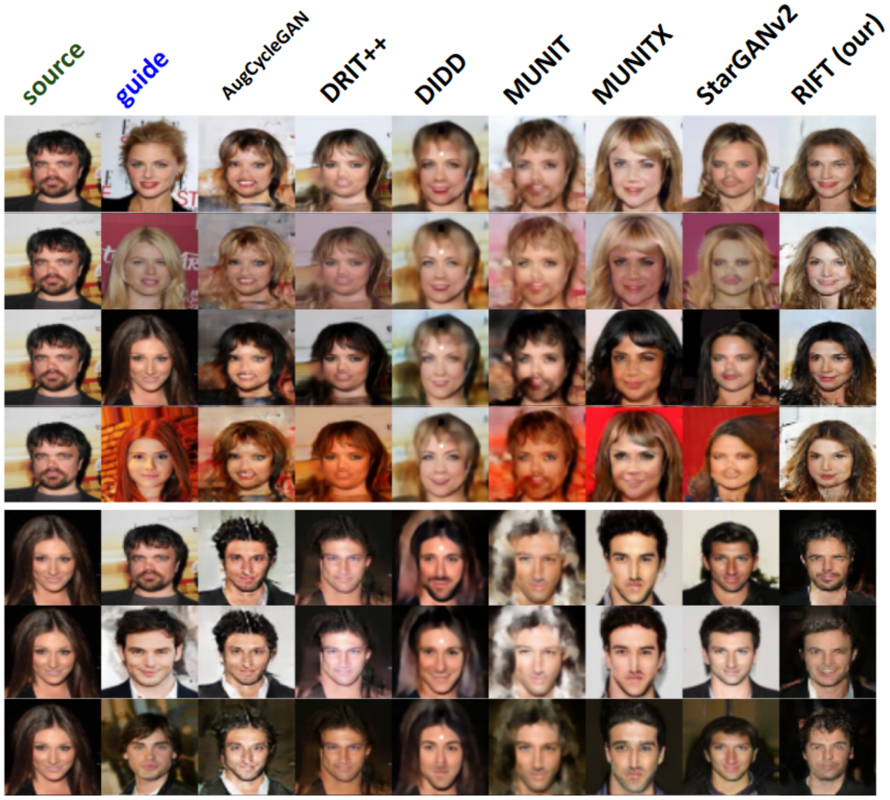


Figure 4·16: Qualitative comparison to prior work on CelebA-FM. Methods should *preserve the pose and the background* of the source, and apply *only the hair color* of the female guide during male2fem translation (top) and *only the facial hair* of the male guide during fem2male translation (bottom). Only RIFT and DIDD preserved background colors *and* applied correct target-specific hair colors and mustaches.

Method	3DS	SA	CA	**AVG**	**RD**
StarGANv2	45	**82**	51	59	97
MUNIT	58	37	53	49	56
MUNITX	33	52	55	47	74
DRIT++	18	24	55	32	20
AugCycleGAN	12	37	40	29	20
DIDD	44	67	**64**	58	35
RIFT (ours)	**88**	78	60	**75**	**6**
RAND	12	24	49	27	9

Table 4.3: Average (AVG↑) manipulation accuracy (ACC) and relative discrepancy (RD↓) across 3D-Shapes-ABC (3DS), SynAction (SA), and CelebA-FM (CA). Notation: **best**, 2nd best.

ulation accuracy across these three datasets.

Qualitative results. Figures 4·14 and 4·15 show that, in most cases, the proposed method successfully preserves domain-invariant content and applies domain-specific attributes from respective domains on 3D-Shapes and SynAction. Figure 4·16 shows that, on CelebA, our method preserves poses and backgrounds, and applies hair color better than other baselines. On 3D-Shapes-A, our method also preserves object color and applies correct size and orientation better than all alternatives (Figure 4·9a). We provide a more detailed side-by-side qualitative comparison of generated images across all baselines and all datasets in the supplementary. We show that RIFT can modulate domain-specific factors of images while keeping them within their original domains (see supplementary of the original paper [121]).

Quantitative results. Tables 4.3 and 4.4 show that across three splits of 3D-Shapes-ABC our method achieves the highest average manipulation accuracy and the lowest relative discrepancy between accuracies of modeling the same attributes as shared and specific. On SynAction, which *matches* the inductive bias of AdaIN-based methods, our method performs on par with StarGANv2 and outperforms all non-AdaIN methods. On CelebA-FM, our method performs on par with DIDD up to a small margin and outperforms other methods. Overall, RIFT achieves best or

Method	3D-Shapes-ABC												SynAct			CelebA-FM					
	FC		WC		OC		SZ		SH		ORI		PS	IDT	BG	HC	FH	GD	ORI	BG	SC
	C	S	C	S	C	S	C	S	C	S	C	S	C	S	S	S	S	S	C	C	C
StarGANv2	0	99	0	99	0	78	5	56	4	99	0	**96**	**96**	**52**	**99**	**76**	15	97	87	11	22
MUNIT	5	94	0	**99**	0	97	**59**	31	96	58	**99**	61	75	28	7	45	7	90	**89**	43	44
MUNITX	1	50	2	55	8	28	12	16	95	21	**99**	7	93	26	37	64	17	75	83	50	43
DRIT++	7	12	9	19	10	10	27	14	7	15	42	51	52	6	13	23	9	96	**89**	67	44
AugCycGAN	10	8	10	9	11	7	17	13	30	13	7	7	90	8	12	16	30	98	12	42	40
DIDD	38	81	29	22	72	18	41	20	87	43	48	34	89	12	**99**	22	**50**	91	78	**89**	56
RIFT	**100**	**100**	**100**	**100**	**100**	**100**	5	**60**	**98**	**100**	97	**96**	89	47	**99**	22	35	**99**	65	83	**57**
RAND	10	10	10	10	10	10	12	19	24	19	6	6	50	11	11	12	31	99	50	50	50

Table 4.4: Manipulation accuracy for shared/common (C) or domain-specific (S) attributes aggregated across **Shapes-3D-ABC**: floor color (FC), wall color (WC), object color (OC), size (SZ), shape (SH), room orientation (ORI); **SynAction**: pose (PS), identity/clothing (IDT), background (BG); **CelebA-FM**: hair color (HC), facial hair (FH), gender (GD), face orientation (ORI), background (BG) and skin color (SC).

second-best (with a small margin) performance in *each* of the three datasets, whereas both runner-ups (DIDD and StarGANv2) perform poorly on at least one of three datasets (DIDD on SynAction, StarGANv2 on CelebA, both on 3D-Shapes), best average accuracy (AVG) across three datasets, and lowest relative discrepancy (RD).

Ablations. During B2A translation on Shapes-3D-A the model trained with all losses uses object color/shape from the source image and floor/wall color from the guide (Fig. 4·15). If we remove the penalty on the capacity of domain-specific embeddings (L_{norm}), the model ignores the source input (Fig. 4·17a-top): it encodes all attributes into domain-specific embeddings, and cycle-reconstructs inputs a and b perfectly from these embeddings (Fig. 4·17a-bottom), completely ignoring the source input: $\boldsymbol{b} = \boldsymbol{F}_{\text{A2B}}(\boldsymbol{a}, \boldsymbol{b}) = b_{\text{cyc}}$. Removing honesty losses (L_{guess}), on the other hand, results in a model that ignores the guide input altogether (Fig. 4·17b-top). The model "hides" domain-specific information inside generated translations instead of the domain-specific embeddings, and makes domain-specific embeddings equal zero, resulting in zero capacity loss $L_{\text{norm}} = 0$, and zero cycle reconstruction loss $L_{\text{cyc}} = 0$. For example (Fig. 4·17b-bottom), the size and orientation of $\boldsymbol{b}$ is hidden inside $\boldsymbol{F}_{\text{B2A}}(\boldsymbol{b}, \boldsymbol{a})$

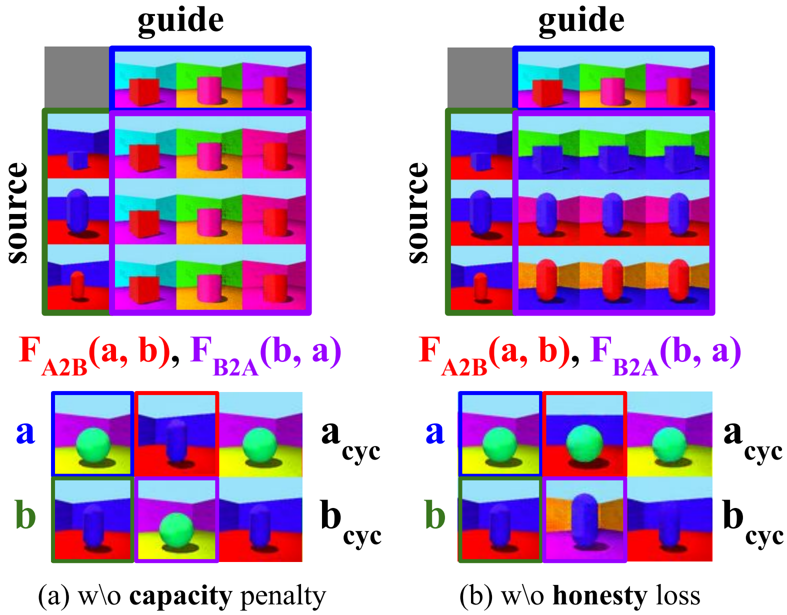

guide
source
$F_{A2B}(a, b)$, $F_{B2A}(b, a)$
a
b
a_{cyc}
b_{cyc}
(a) w\o **capacity** penalty
guide
source
$F_{A2B}(a, b)$, $F_{B2A}(b, a)$
a
b
a_{cyc}
b_{cyc}
(b) w\o **honesty** loss

Figure 4·17: Ablations. Effects of disabling capacity and honesty losses on guided translations (top) and guided cycle-reconstructions (bottom) on Shapes-3D-A. Inputs images from domains **A** and **B**, **A2B** and **B2A** guided translations.

in the form of imperceptible adversarial noise and is used to reconstruct b_{cyc} perfectly. If mapping F_{A2B} actually used size and orientation of b to generate b_{cyc}, it would have also applied that same size and orientation when generating $\boldsymbol{F_{\text{A2B}}(a, b)}$, but it did not - so we conclude that both F_{A2B} and F_{B2A} ignore domain-specific embeddings and embed information inside generated translations instead (see supplementary of the original paper [121]). We also confirmed that the model trained with all proposed losses does *not* hide information inside generated images: we trained a separate classification network to predict attributes of the inputs that *should have been lost* during translation from translated images. The resulting classifier was able to accurately predict hidden information from images generated *without* honesty losses and was unable to predict them (above chance) from images generated by a model trained *with* honesty loss. This confirms that shared attributes of the guide and domain-specific attributes of the source were indeed correctly ignored by the translation network trained with all proposed losses - see supplementary of the original paper [121].

Challenges. We suggest two major causes of remaining errors that existing methods fail to handle at the moment, and future researchers will need to address to make further progress in this task possible. First, some attributes "affect" very different numbers of pixels in training images, and as a consequence contribute very differently to reconstruction losses, making the job of balancing different loss components much harder. For example, the floor color in 3D-Shapes "affects" roughly half of all image pixels, whereas size affects only one-tenth of all pixels - resulting in drastically different effective weights across all losses, especially if both are either domain-specific or shared at the same time. Second, unevenly distributed shared attributes in real-world in-the-wild datasets (such as CelebA) pose an even more serious challenge, rendering the many-to-many problem task *not well defined*. For example, if both male and female domains had hair color variation, but males were mostly brunet with only 3% of blondes, but females were equally likely to be blondes and brunettes - should the model preserve blonde hair when translating females to males and sacrifice the "realism" of the generated male domain, or should it treat hair-color as a domain-specific attribute despite variations present in both? This poses an open question.

Ethical considerations. While more precise attribute manipulation models requiring less supervision might be used for malicious deepfakes [22, 91], they can also be used to remove biases present in existing datasets [40] to promote fairness in downstream tasks [4]. We acknowledge that the CelebA dataset contains many biases (*e.g.* being predominantly white) and that binary gender labels are problematic and encourage the community to collect more inclusive datasets.

Conclusion. In this section we proposed RIFT - a new unsupervised many-to-many image-to-image translation method that determines which factors of variation are shared and which are domain-specific *from data*, and achieves consistently high attribute manipulation accuracy across a wide range of datasets with different kinds of domain-specific and shared attributes, and the low discrepancy between these accuracies. We provide ablations confirming that the self-adversarial embedding takes place in the many-to-many setting and that the honesty loss prevents it from happening. We also show that the capacity loss restricts the effective capacity of the domain-specific embedding in agreement with the provided theoretical bound. Finally, we identified core challenges that need to be resolved to enable further development of unsupervised many-to-many image-to-image translation.

Chapter 5

Future Work

In this chapter, we identify several key future applications of the ideas introduced in this thesis, as well as research directions in which, in our opinion, immediate further progress can be made.

Combating vanishing generator gradients via generalized instance noise. As demonstrated in Section 3.2, likelihood-ratio minimizing flows introduced in this thesis do not suffer from mode collapse and training instability of adversarial alignment methods, but still, experience vanishing of generator gradients in higher dimensions. This failure mode of generative networks was extensively researched in prior work [2, 84, 100], with most authors agreeing that adaptive discriminator regularization is the key, with instance noise being the simplest form of discriminator regularization to implement in practice. One of the most recent and notable is the work of Karras et al. [54], who showed that the discriminator can be regularized with arbitrary image augmentations instead of additive instance noise without any loss of image quality, as long as these augmentations do not make distinct distributions indistinguishable after the transformation. Incorporating these techniques into the framework of likelihood-ratio minimizing flows is the next major step towards being able to apply this stable and predictable method to more complex real-world distributions.

Applying dualization to neural tangent kernels. As demonstrated in Section 3.1, closed-form exact dualization is possible in only a handful of cases. Independently, Li et al. [68] successfully dualized linear approximations of neural discriminators around their optimas with respect to their weight vectors, but the procedure is complicated, preventing more widespread adoption of this technique. Recently, Jacot et al. [51] showed that deep neural networks can be accurately approximated via their linear Taylor expansions around their initial parameter vectors, resulting in accurate predictions of the learning dynamics of resulting networks via so-called neural tangent kernels. Incorporating insights given by the theory of neural tangent kernels into the dualization of discriminators for improved stability is another potentially fruitful direction.

Understanding and improving the stability of self-adversarial defenses. Self-adversarial defenses of Bashkirova et al. [7] were shown to improve the semantic consistency of learned cross-domain mappings, as discussed in Section 3.3, and play a key role in disentangling domain-specific factors from domain-invariant ones, as discussed in Section 4.2.3. We argue that gaining a better understanding of the interplay between adversarial alignment and adversarial self-defense losses, and the stability of the overall procedure is the key to future deployment of these techniques in real-world applications.

Applying cross-domain image manipulation to interpretability. Techniques for controlled manipulation of individual visual attributes of real images using synthetic supervision proposed in Section 4.1 are indispensable for measuring the effect of particular individual factors on the outputs of the downstream model, if counterfactual experiments are economically unfeasible, which is often the case in most important application domains. Moreover, techniques for manipulation of domain-specific attributes in isolation from domain-invariant in the absence of any additional supervision could be very helpful for the identification of biases or factors of variation present in the test dataset that are not present in the training dataset, if we training and deployment datasets come from different distributions.

Bibliography

[1] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented CycleGAN: Learning many-to-many mappings from unpaired data. *arXiv preprint arXiv:1802.10151*, 2018. 84, 85, 97

[2] Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017. 17, 44, 52, 107

[3] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *International Conference on Computer Vision*, pages 4561–4569, 2019. 67

[4] Sean Augenstein, H. Brendan McMahan, Daniel Ramage, Swaroop Ramaswamy, Peter Kairouz, Mingqing Chen, Rajiv Mathews, and Blaise Aguera y Arcas. Generative models for effective ML on private, decentralized datasets. In *International Conference on Learning Representations*, 2020. 105

[5] Kyungjune Baek, Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Hyunjung Shim. Rethinking the truly unsupervised image-to-image translation. In *International Conference on Computer Vision*, pages 14154–14163, October 2021. 85, 97

[6] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018. 33

[7] Dina Bashkirova, Ben Usman, and Kate Saenko. Adversarial self-defense for cycle-consistent GANs. *Advances in Neural Information Processing Systems*, 32, 2019. 8, 9, 53, 57, 80, 85, 91, 108

[8] Dina Bashkirova, Ben Usman, and Kate Saenko. Evaluation of correctness in unsupervised many-to-many image translation. In *Winter Conference on Applications of Computer Vision*, 2022. 9, 85, 86, 89

[9] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. *Advances in Neural Information Processing Systems*, 2007. 27

[10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010. 16

[11] Sagie Benaim, Michael Khaitov, Tomer Galanti, and Lior Wolf. Domain intersection and domain difference. In *International Conference on Computer Vision*, pages 3445–3453, 2019. 84, 85, 97

[12] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. 1, 3

[13] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 2019. 33

[14] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`. 45

[15] Jinming Cao, Oren Katzir, Peng Jiang, Dani Lischinski, Danny Cohen-Or, Changhe Tu, and Yangyan Li. DiDA: Disentangled synthesis for domain adaptation. *arXiv:1805.08019*, 2018. 66, 68, 79, 80

[16] Yun Cao, Zhiming Zhou, Weinan Zhang, and Yong Yu. Unsupervised diverse colorization via generative adversarial networks. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 151–166. Springer, 2017. 12

[17] Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2018. 66

[18] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2016. 66, 86

[19] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 2

[20] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020. 2, 67, 84, 85, 97

[21] Casey Chu, Andrey Zhmoginov, and Mark Sandler. CycleGAN, a master of steganography. *arXiv preprint arXiv:1712.02950*, 2017. 5, 15, 54, 63, 85, 91, 98

[22] Danielle K Citron and Robert Chesney. Disinformation on steroids: The threat of deep fakes. *Cyber Brief*, 2018. 105

[23] Wikimedia Commons. File:ulysses s. grant at city point.jpg — wikimedia commons, the free media repository, 2018. URL https://commons.wikimedia.org/w/index.php?title=File:Ulysses_S._Grant_at_City_Point.jpg&oldid=316583889. [Online; accessed 5-April-2022]. 2

[24] Joshua V. Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matthew D. Hoffman, and Rif A. Saurous. Tensorflow distributions. *CoRR*, abs/1711.10604, 2017. 45

[25] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. 18, 33, 46

[26] Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via Maximum Mean Discrepancy optimization. *31st Conference on Uncertainty in Artificial Intelligence*, 2015. 21

[27] Bernhard Egger, Sandro Schönborn, Andreas Schneider, Adam Kortylewski, Andreas Morel-Forster, Clemens Blumer, and Thomas Vetter. Occlusion-aware 3D morphable models and an illumination prior for face image analysis. *International Journal of Computer Vision*, 2018. 76

[28] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3D morphable face models—past, present, and future. *ACM Transactions on Graphics*, 39(5):1–38, 2020. 2

[29] Geoff French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018. 17

[30] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, 2015. 16

[31] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 17

[32] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 84

[33] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):643–660, 2001. 70, 76

[34] Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International Conference on Machine Learning*, 2013. 21

[35] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 8, 13, 20, 22

[36] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *International Conference on Learning Representations*, 2019. 18, 33, 46

[37] Peter J Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society. Series B*, pages 149–192, 1984. 26

[38] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 7, 14, 20, 21, 25

[39] Aditya Grover, Christopher Chute, Rui Shu, Zhangjie Cao, and Stefano Ermon. Alignflow: Learning from multiple domains via normalizing flows. In *Deep Generative Models for Highly Structured Data Workshop at the International Conference on Learning Representations*, 2019. 33, 35, 45, 47, 50

[40] Aditya Grover, Jiaming Song, Ashish Kapoor, Kenneth Tran, Alekh Agarwal, Eric J Horvitz, and Stefano Ermon. Bias correction of learned generative models using likelihood-free importance weighting. *Advances in Neural Information Processing Systems*, 32, 2019. 105

[41] Ananya Harsh Jha, Saket Anand, Maneesh Singh, and VSR Veeravasarapu. Disentangling factors of variation with cycle-consistent variational auto-encoders. In *European Conference on Computer Vision*, 2018. 66, 67, 79, 81

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 12

[43] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-VAE: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2017. 66, 86

[44] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1989–1998. PMLR, 2018. 17

[45] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, pages 601–608, 2007. 21

[46] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *International Conference on Computer Vision*, pages 1501–1510, 2017. 84, 85, 89

[47] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision*, pages 172–189, 2018. 7, 59, 66, 79, 81, 83, 84, 85, 89, 97

[48] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015. 12

[49] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017. 14, 55, 59, 98

[50] Tommi S Jaakkola and David Haussler. Probabilistic kernel regression models. In *Society for Artificial Intelligence and Statistics*, 1999. 5, 23

[51] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018. 108

[52] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Conference on Computer Vision and Pattern Recognition*, pages 12627–12637, 2019. 83

[53] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 95

[54] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems*, 2020. 17, 107

[55] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 86, 95, 96

[56] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 15

[57] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018. 18, 33, 36

[58] Russell A Kirsch, Lee Cahn, C Ray, and Genevie H Urban. Experiments in processing pictorial information with a digital computer. In *Papers and discussions presented at the December 9-13, 1957, eastern joint computer conference: Computers with deadlines to meet*, pages 221–229, 1957. 1

[59] Adam Kortylewski, Andreas Schneider, Thomas Gerig, Bernhard Egger, Andreas Morel-Forster, and Thomas Vetter. Training deep face recognition systems with synthetic data. *arXiv preprint arXiv:1802.05891*, 2018. 75

[60] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 11, 12, 78

[61] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Conference on Computer Vision and Pattern Recognition*, 2020. 95

[62] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *European Conference on Computer Vision*, 2018. 66

[63] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Kumar Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation viadisentangled representations. *International Journal of Computer Vision*, pages 1–16, 2020. 84, 85, 97

[64] Adrien Marie Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes: avec un supplément contenant divers perfectionnemens de ces méthodes et leur application aux deux comètes de 1805*. Courcier, 1806. 10

[65] Kathleen M Lewis, Srivatsan Varadharajan, and Ira Kemelmacher-Shlizerman. TryOnGAN: Body-aware try-on via layered interpolation. *ACM Transactions on Graphics*, 40(4):1–10, 2021. 83

[66] José Lezama. Overcoming the disentanglement vs reconstruction trade-off via jacobian supervision. In *International Conference on Learning Representations*, 2018. 2, 4, 6

[67] Yujia Li, Kevin Swersky, and Rich Zemel. Generative Moment Matching Networks. *International Conference on Machine Learning*, 2015. 21

[68] Yujia Li, Alexander Schwing, Kuan-Chieh Wang, and Richard Zemel. Dualing gans. In *Advances in Neural Information Processing Systems*, pages 5611–5621, 2017. 21, 107

[69] Seppo Linnainmaa. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16(2):146–160, 1976. 11

[70] Alexander H Liu, Yen-Cheng Liu, Yu-Ying Yeh, and Yu-Chiang Frank Wang. A unified feature disentangler for multi-domain image translation and manipulation. In *Advances in Neural Information Processing Systems*, 2018. 66, 68

[71] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017. 5, 15, 53, 59, 93, 95

[72] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *IEEE International Conference on Computer Vision*, 2019. 84, 85

[73] Yen-Cheng Liu, Yu-Ying Yeh, Tzu-Chien Fu, Sheng-De Wang, Wei-Chen Chiu, and Yu-Chiang Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. In *Computer Vision and Pattern Recognition*, 2018. 66, 68, 80

[74] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, 2015. 68

[75] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision*, December 2015. 87

[76] Mingsheng Long and Jianmin Wang. Learning transferable features with deep adaptation nets. In *International Conference on Machine Learning*, 2015. 16, 31

[77] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2208–2217. JMLR. org, 2017. 16

[78] Matthew M Loper and Michael J Black. OpenDR: An approximate differentiable renderer. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *European Conference on Computer Vision*, 2014. ISBN 978-3-319-10584-0. 66

[79] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Unsupervised learning for real-world super-resolution. In *International Conference on Computer Vision Workshop*, pages 3408–3416. IEEE, 2019. 12

[80] Liqian Ma, Xu Jia, Stamatios Georgoulis, Tinne Tuytelaars, and Luc Van Gool. Exemplar guided unsupervised image-to-image translation with semantic consistency. In *International Conference on Learning Representations*, 2019. 85, 97

[81] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *International Conference on Computer Vision*, 2017. 74, 93, 98

[82] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, 2016. 66, 67

[83] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. 49

[84] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine Learning*, pages 3481–3490. PMLR, 2018. 17, 107

[85] Thomas P Minka. A comparison of numerical optimizers for logistic regression. 2003. 23

[86] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. 17

[87] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019. 36

[88] John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A*, 135(3):370–384, 1972. 10

[89] Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *Society for Industrial and Applied Mathematics Journal on Optimization*, 2004. 22

[90] Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 2009. 22

[91] Thanh Thi Nguyen, Cuong M Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, and Saeid Nahavandi. Deep learning for deepfakes creation and detection: A survey. *arXiv preprint arXiv:1909.11573*, 2019. 105

[92] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. *Advances in Neural Information Processing Systems*, 29, 2016. 21

[93] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision*, pages 269–286, 2018. 98

[94] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 2020. 85

[95] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3D face model for pose and illumination invariant face recognition. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*. Ieee, 2009. 66

[96] H Baden Pritchard. Nicephore niepce. *Nature*, 16(399):142–142, 1877. 1

[97] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems 32*, pages 14680–14691. Curran Associates, Inc., 2019. 36

[98] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015. 5, 17, 33

[99] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118. Springer, 2016. 57, 59

[100] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems*, pages 2018–2028, 2017. 17, 52, 107

[101] Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. In *Conference on Computer Vision and Pattern Recognition Workshop*, June 2018. 98

[102] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. 11

[103] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, 47:3–18, 2016. 70, 75

[104] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Conference on Computer Vision and Pattern Recognition*, 2018. 17

[105] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 22, 31, 59

[106] David W Scott. Multivariate density estimation and visualization. In *Handbook of computational statistics*, pages 549–569. Springer, 2012. 78

[107] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised MAP inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016. 52

[108] Derrick Story. From darkroom to desktop—how photoshop came to light. *Story Photography*, 18, 2000. 1

[109] Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450. Springer, 2016. 16, 33

[110] Ximeng Sun, Huijuan Xu, and Kate Saenko. Twostreamvan: Improving motion modeling in video generation. In *Winter Conference on Applications of Computer Vision*, pages 2744–2753, 2020. 87, 95, 96

[111] Attila Szabó, Qiyang Hu, Tiziano Portenier, Matthias Zwicker, and Paolo Favaro. Challenges in disentangling independent factors of variation. *International Conference on Learning Representations Workshop Track*, 2017. 66, 67

[112] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 5

[113] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015. 33

[114] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016. 2, 66

[115] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014. 16, 22

[116] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *International Conference on Computer Vision*, pages 4068–4076, 2015. 16

[117] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017. 16, 17, 31, 33

[118] Ben Usman, Kate Saenko, and Brian Kulis. Stable distribution alignment using the dual of the adversarial distance. In *International Conference on Learning Representations Workshop (ICLR-W)*, 2018. 8

[119] Ben Usman, Nick Dufour, Kate Saenko, and Chris Bregler. PuppetGAN: Cross-domain image manipulation by demonstration. In *International Conference on Computer Vision*, pages 9450–9458, 2019. 7, 9, 80, 83

[120] Ben Usman, Avneesh Sud, Nick Dufour, and Kate Saenko. Log-likelihood ratio minimizing flows: Towards robust and quantifiable neural distribution alignment. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21118–21129. Curran Associates, Inc., 2020. 8

[121] Ben Usman, Dina Bashkirova, and Kate Saenko. Disentangled unsupervised image translation via restricted information flow. *CoRR*, abs/2111.13279, 2021. 9, 96, 98, 100, 102, 104

[122] Tycho FA van der Ouderaa and Daniel E Worrall. Reversible GANs for memory-efficient image-to-image translation. In *Conference on Computer Vision and Pattern Recognition*, pages 4720–4728, 2019. 36

[123] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015. 12

[124] Wayne Wu, Kaidi Cao, Cheng Li, Chen Qian, and Chen Change Loy. Transgaga: Geometry-aware unsupervised image-to-image translation. In *Conference on Computer Vision and Pattern Recognition*, June 2019. 97

[125] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. PointFlow: 3D point cloud generation with continuous normalizing flows. In *International Conference on Computer Vision*, pages 4541–4550, 2019. 33, 35

[126] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *Conference on Computer Vision and Pattern Recognition*, pages 2528–2535. IEEE, 2010. 13

[127] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision*, 2017. 5, 8, 15, 53, 59, 73, 79, 85, 91

CURRICULUM VITAE

Ben Usman

[redacted]

[redacted]