Batch Weight for Domain Adaptation With Mass Shift

Mikołaj Bińkowski*[1, 2], R Devon Hjelm[1, 3], and Aaron Courville[1, 4]

[1]Mila, Université de Montréal
[2]Imperial College London
[3]Microsoft Research
[4]CIFAR Fellow

Abstract

Unsupervised domain transfer is the task of transferring or translating samples from a source distribution to a different target distribution. Current solutions unsupervised domain transfer often operate on data on which the modes of the distribution are well-matched, for instance have the same frequencies of classes between source and target distributions. However, these models do not perform well when the modes are not well-matched, as would be the case when samples are drawn independently from two different, but related, domains. This mode imbalance is problematic as generative adversarial networks (GANs), a successful approach in this setting, are sensitive to mode frequency, which results in a mismatch of semantics between source samples and generated samples of the target distribution. We propose a principled method of re-weighting training samples to correct for such mass shift between the transferred distributions, which we call batch weight. We also provide rigorous probabilistic setting for domain transfer and new simplified objective for training transfer networks, an alternative to complex, multi-component loss functions used in the current state-of-the art image-to-image translation models. The new objective stems from the discrimination of joint distributions and enforces cycle-consistency in an abstract, high-level, rather than pixel-wise, sense. Lastly, we experimentally show the effectiveness of the proposed methods in several image-to-image translation tasks.

1. Motivation

Recent developments originating from Generative Adversarial Networks [*GANs*, 9] allow generation of high quality images, often hardly distinguishable from the real images [18, 4]. Adversarial methods have also been successfully applied in conditional image generation, where generated samples are obtained as functions of some prior data. When the latter is also composed of images, the problem is also called *image-to-image (domain) transfer* or *style transfer*. In this scope, adversarial objectives are often combined with other loss functions to ensure desirable properties of the transfer networks. These include variants of *cycle consistency* loss, originally proposed in CycleGAN [38] and developed further by [16, 1, 23], leading to current state-of-the-art results in several image-to-image transfer problems.

Notwithstanding these notable results, not much attention has been paid towards understanding of unsupervised domain transfer from probabilistic point of view. Although original CycleGAN assumes deterministic transfer, later works identified the necessity of learning non-deterministic *many-to-many* mappings to account for features that might be present only in one of the considered domains. A common assumption made in such setting, sometimes directly [e.g., 16], but often implicitly, is the existence of latent variable that covers the *shared semantics* between the domains of interest. Learning the transfer function can then be decomposed to learning deterministic encoders to and stochastic decoders from such latent space.

This view, however, does not take into account the distributional differences that may exist between the domains being matched. Since probability mass is preserved trough encoders and decoders, every mode in source domain covers the same share of the distribution as its representations in latent space and the target domain do in their respective distributions. However, we do not necessarily want to match modes that consist the same shares of data.

For example, consider the task of transferring between handwritten digits [*MNIST*, 22], and Street View House Numbers [*SVHN*, 28]. These datasets are independently sampled but share semantics expressed in the digit classes, styless (e.g., seven with or without a cross), etc which we

*Corresponding author (mikbinkowski at gmail dot com).

wish to correctly transfer. Since digits in MNIST are evenly distributed, we expect the *correct* transfer function to produce samples in which zeros cover approximately 10% of all generated ones. Yet, the distribution of the "correct" transfer function (one that maintains the digit class) would be different from the actual SVHN distribution, where ones cover around 20% of the data. Therefore such a transfer function would not be optimal in the Optimal Transport sense, i.e. it would not minimize any divergence between reference and generated distributions. Given such correct transfer-generator, a *good* discriminator would need to be insensitive to the disparity of mode (i.e., digit) frequencies between the source and target distributions. If not, it would provide a gradient signal to the transfer-generator that would encourage it to alter some modes to account for the missing mass of ones in its output.

We will call the described issue a *mode-mass imbalance*.

This issue demonstrates the view that shared semantics can be modelled through a latent variable is, in general, invalid. However, the described problem is inherent in all GAN-based approaches to domain transfer, since GAN discriminators are always trained to estimate some kind of divergence e.g., *Jensen-Shannon* [9]; *Wasserstein distance* [2, 11]; *Maximum Mean Discrepancy* [24, 3] between reference distribution and the generated samples. For these reasons, we propose *batch-weight* as a solution to the issue caused by mass-preserving property of optimal transport in the context of domain transfer. Batch weight aims to re-balance samples within each batch to account for differences in the reference and generated distributions.

2. Related Work

The need to correct for mode shift between distributions of interest has been widely studied in machine learning.

2.1. Supervised learning

In supervised learning, it is often assumed that distributions $p(x)$ and $q(x)$ of the independent variable on the training and tests are different, but the conditionals $p(y|x)$ and $q(y|x)$ are equal. Such situation is known as *covariate shift* or *sample selection bias* and has been addressed in multiple works [32, 36, 15, 10, 33].

The complimentary setting when $p(x|y) = q(x|y)$, termed *label shift*, has also been studied [37]. In more recent work, [25] consider label shift correction for black box predictors.

2.2. Importance sampling

The problem of changing probability measure in empirical setting has has long been studied in the field of *importance sampling*. This general technique has been used in estimating properties of distribution available indirectly through another distribution.

Importance sampling is often applied in variance reduction problems. In such, one re-weights the available sample so that the variance of the estimated quantity under new distribution is lower than with respect to the original one.

2.3. Domain transfer

The distribution shift has undergone some limited study in the context of domain transfer. [5] empirically showed that the use of distribution-matching loss functions in domain transfer leads to issues when modes in target domain are under- or over-represented as compared to the source. [6] also identifies the problem of distribution mismatch in generative modelling, however the proposed re-balancing function is provided only in case when relation between source and target distribution is available (directly or indirectly).

[35] proposed an algorithm based on Optimal Transport for distribution matching within a single domain, which is then applied to image-to-image transfer by using a single *Variational Autoencoder* [*VAE*, 20] trained on the union of available samples. It therefore depends on the quality of reconstructions and feasibility of encoding two different domains using a single VAE. As a two-step method, it does not directly optimize the transfer and adjustment of the imbalance between the original samples, but between their embeddings.

State-of-the-art unsupervised image-to-image translation models [16, 1, 23] assume that we are given samples X and Y drawn from two domains $\mathcal{X}$ and $\mathcal{Y}$ according to some distributions $\mathbb{P}_x$ and $\mathbb{Q}_y$, and seek (possibly random) generator functions $G_{xy} : \mathcal{X} \to \mathcal{Y}$ and $G_{yx} : \mathcal{Y} \to \mathcal{X}$ so that generated distributions $G_{xy}\#\mathbb{P}_x$ and $G_{yx}\#\mathbb{Q}_y$[1] match with $\mathbb{Q}_y$ and $\mathbb{P}_x$ in *consistent* way. Numerous techniques have been developed to ensure that the generated samples are consistent with their sources and that the transfer is invertible. Most of them are based on simple yet powerful *cycle-consistency loss* [38],

$$L^{cyc}(x, y) = \|G_{yx}(G_{xy}(x)) - x\|_1 + \|G_{xy}(G_{yx}(y)) - y\|_1, \quad (1)$$

along with the GAN objective [9]; [16, 23, 39] combine as many as five different loss components to train the generator functions.

Note that in the presence of mode-mass imbalance, the assumption that generators should minimize the GAN objective violates the consistency between sources and targets. It can be shown that this assumption together with cycle consistency imply that our samples are drawn from marginals of the same distribution $\mathbb{P}_{xy} = \mathbb{Q}_{xy}$ on $\mathcal{X} \times \mathcal{Y}$, i.e. $\mathbb{P}_x = \mathbb{E}_y\mathbb{P}_{xy}, \mathbb{Q}_y = \mathbb{E}_x\mathbb{P}_{xy}$.

[1] $f\#\mathbb{P}$ denotes *push-forward* measure of $\mathbb{P}$ through function f.

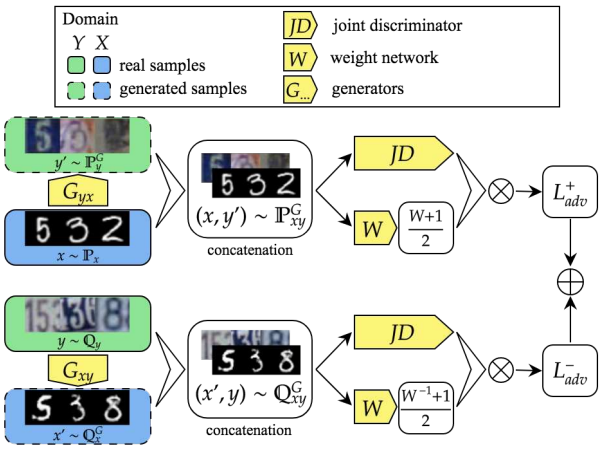


Figure 1: Scheme of the proposed model. Joint distributions are matched adversarially, with the only supervision coming from marginals.

More complicated loss functions used in domain transfer still seek the generators that mimic the conditionals $\mathbb{P}_{y|x}$ and $\mathbb{Q}_{x|y}$, given marginals $\mathbb{P}_x$ and $\mathbb{Q}_y$. Although they do not imply the equality of the joint distributions $\mathbb{P}_{xy}, \mathbb{Q}_{xy}$ being searched (which correspond to the correct transfer), they do imply equality of their marginals, $\mathbb{P}_x = \mathbb{Q}_x$ and $\mathbb{P}_y = \mathbb{Q}_y$, which is impossible in the presence of mode-mass imbalance.

Domain transfer models often assume existence of underlying *shared semantics* [e.g. 16] modelled as underlying latent variable U that consists common features of X and Y. In such scenario, one aims to train encoders $Enc(U|X), Enc(U|Y)$ and decoders $Dec(X|U), Dec(Y|U)$, and transfer between domains through U. This, however, does not account for the possible mode-mass imbalance between X and Y: even non-deterministic encoders and decoders preserve probability mass between X and U, and U and Y. Therefore, the assumption that such U exists is, in general, invalid. This inherent issue of domain transfer has been noted by [21].

3. Formulation

In this section we propose a framework to perform unsupervised domain transfer in the presence of mode-mass imbalance, without cycle-consistency loss.

Assume that $\mathbb{P}_{xy}, \mathbb{Q}_{xy}$ are distributions on $\mathcal{X} \times \mathcal{Y}$ such that supp $\mathbb{P}_{xy} =$ supp $\mathbb{Q}_{xy}$ and that we observe samples drawn from their marginals $\mathbb{P}_x$ and $\mathbb{Q}_y$. $\mathbb{P}_{xy}$ and $\mathbb{Q}_{xy}$ represent correct matchings between these domains, i.e the transfer from x to y (or other way around) is considered valid if the pair (x, y) can be drawn from $\mathbb{P}_{xy}$ and $\mathbb{Q}_{xy}$. Although we do not assume the same marginals,we do assume the equality of conditionals,

$$\mathbb{P}_{y|x} = \mathbb{Q}_{y|x} \qquad \mathbb{P}_{x|y} = \mathbb{Q}_{x|y}. \tag{2}$$

This assumption is much weaker than equality of joints $\mathbb{Q}_{xy} = \mathbb{P}_{xy}$ which most domain transfer models implicitly assume.

We aim to obtain the correct transfer by learning generators $G_{xy} : \mathcal{X} \to \mathcal{Y}, G_{yx} : \mathcal{Y} \to \mathcal{X}$ that mimic the above marginals. Let $\mathbb{P}^G_y = G_{xy}\#\mathbb{P}_x$ and $\mathbb{Q}^G_x = G_{yx}\#\mathbb{Q}_y$.

Let $\mathbb{M} = \frac{1}{2}(\mathbb{P}_{xy} + \mathbb{Q}_{xy})$. Thanks to the assumption of equality of the supports of $\mathbb{P}_{xy}$ and $\mathbb{Q}_{xy}$, both Radon-Nikodym derivatives $w = \frac{d\mathbb{P}_{xy}}{d\mathbb{Q}_{xy}}$ and $v = \frac{d\mathbb{Q}_{xy}}{d\mathbb{P}_{xy}}$ exist and satisfy,

$$w(x, y) = (v(x, y))^{-1}, \qquad (x, y) \in \text{supp } \mathbb{P}_{xy}. \tag{3}$$

Therefore

$$\begin{aligned}\mathbb{P}_{xy}\tfrac{1}{2}(1 + w(X, Y)) &= \mathbb{M} \\ &= \mathbb{Q}_{xy}\tfrac{1}{2}(1 + w^{-1}(X, Y)).\end{aligned} \tag{4}$$

As we aim to learn the distributions $\mathbb{P}_{xy}$ and $\mathbb{Q}_{xy}$ through generators G_{xy} and G_{yx}[2],

$$\begin{aligned}\mathbb{P}_{xy} \approx \mathbb{P}^G_{xy} &:= (\text{id} \otimes G_{yx})\#\mathbb{P}_x, \\ \mathbb{Q}_{xy} \approx \mathbb{Q}^G_{xy} &:= (G_{xy} \otimes \text{id})\#\mathbb{Q}_y.\end{aligned}$$

We can approximate distributions on the left and right hand sides of the Eq. 4 using available samples from marginals $\mathbb{P}_x$ and $\mathbb{Q}_y$, along with weighting network $W \in \mathcal{W}$ that approximates the derivative $w = \frac{d\mathbb{P}_{xy}}{d\mathbb{Q}_{xy}}$, where $\mathcal{W} = \{W : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+, \mathbb{E}_{X\sim\mathbb{P}_x}[W(X)] = 1\}$. Such constraint can easily be enforced by a softmax layer computed over samples in the batch.

Therefore, at generation step we shall optimize the following objective:

$$\inf_{G_{xy}, G_{yx}, W} \mathbb{E}_{\substack{X\sim\mathbb{P}_x \\ Y\sim\mathbb{Q}_y}} \mathcal{L}\left(\mathbb{P}^G_{xy}\tfrac{1}{2}(1 + W), \mathbb{Q}^G_{xy}\tfrac{1}{2}(1 + W^{-1})\right), \tag{5}$$

where $\mathcal{L}$ is some loss function trained adversarially.

At this point, we introduce a *joint discriminator* $D : \mathcal{X} \times \mathcal{Y} \to R$, a neural network that discriminates between distributions supported on $\mathcal{X} \times \mathcal{Y}$ and R is a domain that depends on the GAN type. Similar idea has been applied in ALI/BiGAN [8, 7]. Joint discriminator enforces cycle-consistency on the abstract- rather that pixel-level as objective 1 does.

Assuming Wasserstein GAN setting and $R = \mathbb{R}$, the full objective implied by Eq. 5 is as follows:

$$\begin{aligned}&\inf_{G_{xy}, G_{yx}, W} \sup_D \Big(\\ &\mathbb{E}_{X\sim\mathbb{P}_x}\tfrac{1}{2}D(X, G_{yx}(X)) \times (1 + W(X, G_{yx}(X))) \\ -&\mathbb{E}_{Y\sim\mathbb{Q}_y}\tfrac{1}{2}D(G_{xy}(Y), Y) \times (1 + W(G_{xy}(Y), Y)^{-1})\Big).\end{aligned} \tag{6}$$

[2] implicitly, by learning the conditionals

The batch weight procedure for Wasserstein domain transfer with joint discriminator is detailed in Algorithm 1. Overview of the algorithm is also shown in Figure 1. From now on, we will call the proposed architecture a *Joint Discriminator - Batch Weighted* domain transfer or shortly *JD-BW*.

Algorithm 1 Batch Weight

Given: $\mathbb{P}_x$ and $\mathbb{Q}_y$ - source and target distributions
Given: d - number of discriminator steps per generator step, N - total training steps, m - batch size
Initialize generators G_{xy}, G_{yx}, discriminator D and weighting W network parameters $\theta_G, \theta_D, \theta_W$.
for $k = 1$ **to** n **do**
  `# generator - weight step`
  Sample $x_1, \dots, x_m \sim \mathbb{P}$ and $y_1, \dots, y_m \sim \mathbb{Q}$.
  $w_1, \dots, w_m \leftarrow \sigma([W(x_i, G_{yx}(x_i))]_{i=1}^m)$
  $v_1, \dots, v_m \leftarrow \sigma(-[W(G_{xy}(y_i), y_i)]_{i=1}^m)$
  $L^- \leftarrow \sum_{i=1}^m D(x_i, G_{xy}(x_i)) \cdot \frac{1}{2}(1 + w_i)$
  $L^+ \leftarrow \sum_{i=1}^m D(G_{yx}(y_i), y_i) \cdot \frac{1}{2}(1 + v_i)$
  $\theta_G \leftarrow \text{Adam}\left(\nabla_G [L^- - L^+], \theta_G\right)$
  $\theta_W \leftarrow \text{Adam}\left(\nabla_W \left[(L^- - L^+)^2\right], \theta_W\right)$
  for $j = 1$ **to** d **do**
    Sample $x_1, \dots, x_m \sim \mathbb{P}$ and $y_1, \dots, y_m \sim \mathbb{Q}$.
    $w_1, \dots, w_m \leftarrow \sigma([W(x_i, G_{yx}(x_i))]_{i=1}^m)$
    $v_1, \dots, v_m \leftarrow \sigma(-[W(G_{xy}(y_i), y_i)]_{i=1}^m)$
    $L^- \leftarrow \sum_{i=1}^m D(x_i, G_{xy}(x_i)) \cdot \frac{1}{2}(1 + w_i)$
    $L^+ \leftarrow \sum_{i=1}^m D(G_{yx}(y_i), y_i) \cdot \frac{1}{2}(1 + v_i)$
    $\theta_D \leftarrow \text{Adam}\left(-\nabla_D [L^- - L^+], \theta_D\right)$
  end for
end for

3.1. Two-domain vs. single-domain batch weight

At early stage of this work we considered one-sided batch weight, where weighting network was applied only to samples coming from one of the domains. That approach, however, proved somewhat unstable: one-sided re-weighting carries a risk of some training examples from the weighted domain getting weights very close to zero, which slows down the training (as they would have small impact on gradients). Some examples may need to be assigned very small weights, yet this can occur wrongly at early phase of training, when weighting network is imperfect. For instance, a possible failure mode would be when such low weights were assigned to the samples of lowest quality (at some point during training); in such case generator would have very little incentive to improve them.

The advantage of re-weighting both domains is that it ensures that every example in each training set would receive a weight no smaller than a half of what it would get without re-weighting. Therefore it may never collapse in the sense that some examples from one domain are fully excluded from training, i.e. assigned zero weights.

In practice, we found re-weighting both domains much more stable. Thanks to the symmetric formulation[3], if some mode gets lower weights in one domain, the corresponding mode in the other domain is likely to receive higher weights, eventually leading to balance between re-weighted domains.

The original idea of one-sided batch-weight is presented in Appendix A in supplementary material.

3.2. Non-uniqueness and implicit bias

Given empirical distributions $\mathbb{P}_x$ and $\mathbb{Q}_y$ there exist many joints $\mathbb{P}_{xy}$ and $\mathbb{Q}_{xy}$ satisfying equality of conditionals (Assumption 2). For instance, $\mathbb{P}^0_{xy} := \mathbb{Q}^0_{xy} := \mathbb{P}_x \otimes \mathbb{Q}_y$ is a valid distribution on $\mathcal{X} \times \mathcal{Y}$ that has all the assumed properties, except that it leads to *independent* transfer between these domains. Mathematically speaking, the problem of domain adaptation is *ill-posed*.

For this reason, it is worth to note the role of implicit bias of the generator network architectures used in modelling the conditional distributions $\mathbb{P}_{y|x}$ and $\mathbb{Q}_{x|y}$. State-of-the art image-to-image translation models [38, 16, 23] all use deep ResNets [13] or U-Nets [30] as transfer-generators. These architectures bias the generator mapping towards identity in pixel-level space, which helps obtaining satisfying transfer networks. This, however, also explains limitations of these models: the most impressive performance has so far been achieved in tasks with near pixel-to-pixel correspondence, also referred to as *style transfer*.

Nevertheless, we follow these approaches and impose similar architectural constraints on G_{xy}, G_{yx} to enforce the dependence structure in learned joints $\mathbb{P}^G_{xy}$ and $\mathbb{Q}^G_{xy}$.

4. Experiments

In this section we discuss experiments, benchmarks and ablation studies for the proposed JD-BW algorithm.

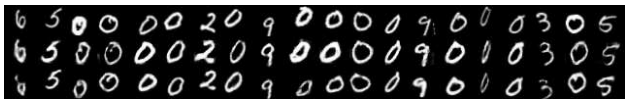
Although at early stage of this work we carried out experiments with one-sided batch-weight mentioned in Section 3.1, they confirmed the aforementioned issues. For this reason, we focus on experiments with Algorithm 1.

4.1. Datasets

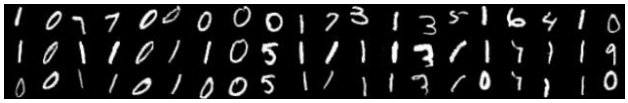
We carry out experiments on four dataset pairs.

1. **MNIST to skewed & resized MNIST**. In this experiment we alter the standard MNIST dataset by introducing bias towards zeros. In the *SR-MNIST* (skewed and resized MNIST) half of the samples are drawn from the class of zeros, while the other half are drawn with equal probabilities from the remaining digit classes. The images are then padded, randomly rotated by the

[3] more precisely, the invertibility of Random-Nikodym derivative.



(a) Joint Discriminator with Batch Weight



(b) MUNIT with Batch Weight

(c) Joint Discriminator alone

(d) MUNIT

Figure 2: Results for SR-MNIST to MNIST transfer. Each picture shows three rows of images: original SR-MNIST samples in the first one, samples transferred to MNIST space in the second one, and 2nd row samples transferred back to SR-MNIST space in the last row. Only Joint Discriminator architecture with Batch weight achieves satisfying results, while all other models struggle with frequency of zeros in SR-MNIST dataset, which are often matched with other digits. Note that JD models are do not directly optimize the quality of reconstructions (2nd rows), whereas MUNIT does so via cycle-consistency.

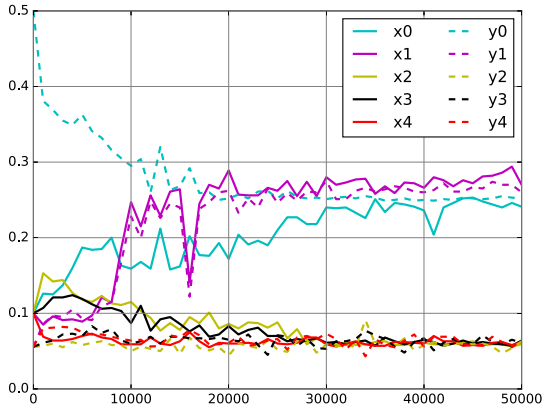


Figure 3: Moving averages of the combined batch weights assigned to each of the digit-classes throughout MNIST to SR-MNIST transfer training; iterations in horizontal axis. For clarity we show values only for first five digits; xk's - solid lines (yk's - dashed lines) stand for digits k in MNIST (SR-MNIST). Even though zeros in SR-MNIST are very frequent, their weights gradually become lower, while those of their MNIST - counterparts - higher, eventually matching with each other. Total weights of other digits match too, with ones getting high weights in both datasets.

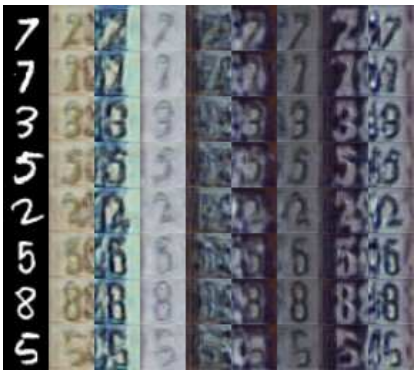


Figure 4: MNIST to SVHN transfer with JD-BW architecture and fixed noise values. Original MNIST samples on the left; samples in each other column were obtained with the same noise sampled from $\mathbb{R}^{16}$.

angle $\alpha \in (-\frac{\pi}{12}, \frac{\pi}{12})$ and randomly cropped. The resulting digits are slightly smaller than the original ones and not necessarily centered. Although changing sampling frequency itself makes the case for batch weighting, the alterations made to the digits so that the transfer to be learnt is not trivial/deterministic.

2. **MNIST to SVHN**. We attempt the transfer between MNIST to SVHN [28] without using the labels. SVHN has a non-uniform distribution of digits and is characterized by considerably more complex features, such us font, colour, background and size. Although we use the version of SVHN with centered digits, they also often contain side-digits coming from the whole house number. To our best knowledge, this problem has not yet been solved.

3. **Edges to Shoes&Bags**. We combine *edges2shoes* and *edges2handbags* datasets [17] to obtain two-class datasets of edges and photos, and alter sampling of the latter so that large share (50%, 70 or 90%) of examples are photos of shoes. In the edge-domain we leave sampling unchanged, hence 50k out of total 188k (26%) examples are contours of shoes. We carry out experiments at 128x128 resolution.

4. **CelebA to Portraits**. We transfer CelebA dataset of celebrity photos [26] to WikiArt dataset of 1714 portraits [23]. We randomly crop images around the faces and resize to 128x128 resolution.

4.2. Benchmarks and ablation study

We compare the performance of the proposed model with MUNIT [16], which is one of the state-of-the-art models in unsupervised image-to-image transfer. We do not compare with CycleGAN [38] and BiCycleGAN [39] as the former does not allow multimodal transfer and MUNIT is essentially its extension, while the latter requires paired training examples.

Since our model has two novel components, batch weight and joint discriminator, we also carry out two ablations on the MNIST - SR-MNIST task (Section 5.1):

- MUNIT with batch weight,
- Joint discriminator architecture without batch weight.

To make our research reproducible, the code for experiments is attached to the submission.

4.3. Evaluation

We propose two quantitative metrics that measure the quality of transfer.

1. *Transferred Samples Accuracy (TSAcurracy)*; the accuracy of the classifier trained on the target domain evaluated on transferred samples with labels from respective source images.
2. *Joint Fréchet Inception Distance (JointFID)*; a metric analogous to *FID* [14] with the difference that the Fréchet distance is computed on joint (concatenated) *Inception*[34][4] representations of source and transferred images.

Both of these metrics measure sample fidelity and correctness of the transfer[5]. JointFID additionally measures sample diversity per domain, as FID does.

TSAccuracy can only be computed if some kinds of modes are known (e.g. if class labels are available), while JointFID requires paired ground-truth samples. For these reasons, we carry out evaluation on MNIST - SR-MNIST and Edges - Shoes&Bags tasks.

Details.
We trained LeNet classifiers for MNIST and SRMNIST datasets and simple DCGAN-discriminator-like classifier for Edges and Shoes&Bags datasets. All of these classifiers are trained for 10,000 steps, using balanced samples and batch size of 128. At evaluation we sample datasets as in the original transfer tasks. We collect 50,000 individual samples for calculation of JointFID and TSAccuracy.

Results are presented in Table 1.

[4]For MNIST - SR-MNIST task we follow [3] and use *LeNet* instead, as pre-trained Inception network is not very expressive for these domains. The same LeNet instances are used for JointFID and TSAccuracy.

[5]For instance, a classifier should not recognize samples of poor quality and, at the same time, should detect the wrongly transferred ones, regardless of their quality.

4.4. Network architectures

Generators
As stressed in Section 3.2, the architecture plays very important role in domain transfer. Following the successful architectures of [38, 16] we use generators with several residual blocks [13] to bias the transfer towards identity.

Since we consider non-deterministic transfer, generator networks take as inputs the image and noise vector, sampled uniformly from $\mathbb{R}^d$, where $d = 8$ for MNIST - SR-MNIST task and $d = 16$ for other tasks. Noise vector is repeated over the spacial dimensions and concatenated to convolutional representation: halfway through the depth of the network for 32x32 models and before the first residual block for 128x128 architectures. We follow [4] and use spectral normalization [27] in both generator and discriminator networks. 32x32 architecture is shown in details in Table 3 in Appendix B in supplementary material. For 128x128 resolution we used the same generators as in MUNIT [16][6].

Discriminator
For joint discriminator, we use somewhat more powerful discriminator than the DCGAN [29], as it has to discriminate between joint distributions on $\mathcal{X} \times \mathcal{Y}$. The architecture at each level separately computes features of each of the images alone and of their concatenation. We use spectral normalization [27] and gradient penalty at training points[7] as in [31]. Details are shown in Table 2 in Appendix B in supplementary material.

Weighting network
For weighting network, we considered several approaches. Function W maps from $\mathcal{X} \times \mathcal{Y}$, yet the samples it will ever see during the training are either of the form $(x, G_{yx}(x)), x \sim \mathbb{P}_x$ or $(G_{xy}(y), y), y \sim \mathbb{Q}_y$; there are thus multiple ways of modelling W. Overall, out of the several strategies we considered for obtaining weights $w_{\mathbf{x}}, w_{\mathbf{y}}$ for the batches $\mathbf{x} \sim \mathbb{P}_x^n, \mathbf{y} \sim \mathbb{Q}_y^n$ the following proved most stable:

$$
\begin{aligned}
&W_x : \mathcal{X} \to \mathbb{R}, \quad W_y : \mathcal{Y} \to \mathbb{R}, \\
&w_{\mathbf{x}} = \tfrac{1}{2}\left(\sigma(W_x(\mathbf{x})) + \sigma(-W_y(G_{xy}(\mathbf{x}))\right), \\
&w_{\mathbf{y}} = \tfrac{1}{2}\left(\sigma(-W_x(G_{yx}(\mathbf{y})) + \sigma(W_y(\mathbf{y}))\right),
\end{aligned}
$$

where the weight networks W_x, W_y are modeled using the architecture of DCGAN discriminator [29] with four convolutional layers and 64 features in the first layer.

We also found useful regularizing the weighting network training by clipping the values of W, which lets us control (and gradually relax) the ratio between highest and lowest weights within a batch. We discuss these further in Appendix B.1 in supplementary material.

[6]except that we used spectral normalization and did not use *Adaptive Instance Norm*.

[7]instead of interpolations between training and reference samples as originally proposed by [12]

4.5. Training details

We train all models using Adam optimizer [19] with parameters $\beta_1 = .5, \beta_2 = .999$, but we used different hyperparameter settings for 32x32 and 128x128 architectures.

32x32 models. We used batch size of 128. Joint Discriminator models are trained with 5 discriminator steps per generator step, while MUNIT (benchmark) models are trained with one generator per one discriminator step, as in original implementation. For these reason, we train the latter for 3x more generator steps than the proposed architectures[8]. After this number of iterations, MUNIT models seemed to converge. Overall, we train JD (MUNIT) for 50k (150k) steps in MNIST - SR-MNIST task and 250k (750k) steps in MNIST - SVHN task.

128x128 models. We used batch size of 6, 2 discriminator steps per one generator step and trained for 300k generator steps.

5. Results

5.1. MNIST to SRMNIST

We perform the unsupervised transfer task from SR-MNIST to MNIST (i.e., without using the labels in any part of our objective for any network). As zeros are over-represented in the first dataset, we anticipate this task will be difficult without properly reweighting the GAN objective. We compare to Multimodal Unsupervised Image-to-Image Translation [*MUNIT*, 16]

Results from this experiment are shown in Figure 2. Only Joint Discriminator architecture with Batch Weight performed well, correctly matching different digit classes. Other models often incorrectly match some kinds of SR-MNIST zeros with other MNIST digits. Quantitive evaluation presented in Table 1 confirms this observation, as JD-BW outperforms MUNIT in both TSAccuracy and JointFID. We monitored the batch weights assigned to each example within a batch in order to find out if weighting network(s) are capable of matching frequencies of the modes in two distributions. Figure 3 presents evolution of the weights aggregated for each of the MNIST/SR-MNIST classes (as these are the only modes we can clearly distinguish). Weighting network successfully allows the mode frequencies to gradually match between both distributions.

We also note that the proposed model achieves cycle-consistency in an abstract, high-level sense. The learned transfer is non-deterministic, yet the reconstructed SR-MNIST samples $G_{xy}(G_{yx}(y))$ belong to the same submanifold as the original samples y, where such submanifold is spanned by features non-existent in MNIST space.

[8]This required slightly longer training for MUNIT anyway, due to simplified JD architecture.

Task	if	TSAccuarcy (%) *higher is better* JD-BW	TSAccuarcy (%) *higher is better* MUNIT	JointFID *lower is better* JD-BW	JointFID *lower is better* MUNIT
MNIST → SR-MNIST	.4	**93**	69	**6.1**	34.7
paired real data	-	98		.02	
MNIST ← SR-MNIST	.4	**93**	44	**6.0**	122.9
paired real data	-	98		.08	
Edges → Shoes&Bags	.5	**97**	96	43	**32.7**
	.7	**97**	76	46.5	**44.7**
	.9	**95**	42	**52**	100.6
paired real data	-	98		11.5	
Edges ← Shoes&Bags	.5	**95**	91	25.3	**21.7**
	.7	**96**	85	**25.7**	29.3
	.9	**91**	80	**33**	69.5
paired real data	-	97		8.7	

Table 1: Quantitative results. *if* - imbalance factor, for MNIST - SR-MNIST tasks it denotes the share of zeros in SR-MNIST, for Edges - Shoes&Bags it denotes the share of photos of shoes in the latter dataset. JD-BW (proposed) outperforms MUNIT when imbalance is present.

5.2. MNIST to SVHN

Results from this experiment are shown in the Figure 5. Although our model does not produce as sharp images as expected and makes few mismatch errors, it provides reasonable transfer. MUNIT, on the other hand, wrongly transfers MNIST to SVHN and collapses completely in the opposite direction.

Our method also disentangles SVHN-specific features from those shared with MNIST, and models the former via noise input. Figure 4 shows samples obtained from different MNIST digits with the same noise, for 8 different noise values.

5.3. Edges to Shoes&Bags

In this experiment the main difficulty comes from difference in frequencies of bags and shoes between source and target domains. As shown in Fig. 6a, the proposed model successfully tackled the problem, producing correctly transferred samples. MUNIT (Fig. 6c), on the other hand, struggled with the imbalance and often transferred bag-edges to shoes, which were over-represented in the photo domain.

Table 1 shows evaluation metrics for this task with varying distribution skew in Shoes&Bags domain. JD-BW obtained higher TSAccurracy than MUNIT regardless of that factor and better JointFID when the imbalance was higher.

5.4. CelebA to Portraits

We present results from this experiment in Fig. 6b and 6d. In this experiment we were not able to quantify mass shift between different modes, as no labels/additional features are available for the Portrait dataset. However, some of such imbalances are visible, e.g. gender proportions and

(a) JD-BW, MNIST to SVHN

(b) JD-BW, SVHN to MNIST

(c) MUNIT, MNIST to SVHN

(d) MUNIT, SVHN to MNIST

Figure 5: Results for MNIST to SVHN transfer. First row in each picture shows original samples, while second the transferred samples. For the proposed JD-BW model, the additional 3rd row presents samples transferred back to the original space. Although MUNIT produces sharper images, the match between original and transferred samples is very poor, with SVHN to MNIST direction suffering from mode collapse. Joint Discriminator struggles with image sharpness, however the matching is usually correct and cycle-consistent.

(a) JD-BW (ours), Edges to Shoes&Bags

(b) JD-BW (ours), Portraits to CelebA

(c) MUNIT, Edges to Shoes&Bags

(d) MUNIT, Portraits to CelebA

Figure 6: Results for Portraits to CelebA and Edges to Shoes&Bags transfers. Original samples are shown in the first row, second one shows samples transferred to the other domain while the third shows 2nd row transferred back to the original domain. For both datasets our model yields very reasonable transfer. MUNIT, on the other hand, struggles to match domains correctly, despite overall good quality of produced samples.

frequency of moustache are different in paintings than in celebrity photos.

Both models produced samples of good quality, however those coming from our model preserved much more features of the original examples. MUNIT, in fact, preserves only a pose, while all other features seem to be independent of the source image. In Appendix C in supplementary material we further discuss the nature of transfer learned by MUNIT in this and in the previous task.

6. Conclusion

In this work, we considered unsupervised domain transfer in the presence of mode imbalance, a situation when modes to be matched have different frequencies in source and target domains. The contributions of this paper are as follows. Firstly, we provide probabilistic formalism of unsupervised domain transfer. Secondly, we propose a novel method of batch weighting to tackle the issue of mode imbalance. Thirdly, we propose a new architecture called Joint Discriminator, that not only largely simplifies the training objective, but also ensures cycle-consistency in multimodal, high-level sense, without directly enforcing quality of reconstructions. Finally, we propose two quantitative evaluation metrics for domain adaptation, Transferred Samples Accuracy and JointFID. We experimentally show effectiveness of our model and its superiority over the existing benchmark in tasks where mode-mass imbalance is present.

Acknowledgements

We would like to thank Arthur Gretton, Dougal J. Sutherland (UCL) and Samuel Lavoie-Marchildon (Mila) for valuable comments and discussions. M.B. received partial support from Engineering and Physical Sciences Research Council (EPSRC). NVIDIA donated a DGX-1 computer used in this work.

References

[1] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron C. Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. *CoRR*, abs/1802.10151, 2018.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[3] Mikołaj Bińkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *CoRR*, abs/1809.11096, 2018.

[5] Joseph Paul Cohen, Margaux Luck, and Sina Honari. Distribution matching losses can hallucinate features in medical image translation. In *MICCAI*, 2018.

[6] Maurice Diesendruck, Ethan R Elenberg, Rajat Sen, Guy W Cole, Sanjay Shakkottai, and Sinead A Williamson. Importance weighted generative networks. *arXiv preprint arXiv:1806.02512*, 2018.

[7] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *CoRR*, abs/1605.09782, 2016.

[8] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[9] 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*, pages 2672–2680, 2014.

[10] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Dataset shift in machine learning. In *Covariate Shift and Local Learning by Distribution Matching*, pages 131–160. MIT Press, 2008.

[11] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved Training of Wasserstein GANs, May 2017.

[12] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, Dec. 2015.

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017.

[15] Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Scholkopf. Correcting sample selection bias by unlabeled data. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 601–608, Cambridge, MA, USA, 2006. MIT Press.

[16] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017.

[19] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. Jan. 2015.

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

[21] Samuel Lavoie-Marchildon, Sebastien Lachapelle, Mikołaj Bińkowski, Aaron Courville, Yoshua Bengio, and R Devon Hjelm. Unsupervised one-to-many image translation. 2018.

[22] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[23] 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.

[24] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pages 2203–2213, 2017.

[25] Zachary C Lipton, Yu-Xiang Wang, and Alex Smola. Detecting and correcting for label shift with black box predictors. *arXiv preprint arXiv:1802.03916*, 2018.

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, Dec. 2015.

[27] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *CoRR*, abs/1802.05957, 2018.

[28] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[29] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[31] Kevin Roth, Aurélien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. *CoRR*, abs/1705.09367, 2017.

[32] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244, 10 2000.

[33] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information pro-*

cessing systems, pages 1433–1440, 2008.

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[35] Karren D. Yang and Caroline Uhler. Scalable unbalanced optimal transport using generative adversarial networks. *CoRR*, abs/1810.11447, 2018.

[36] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 114–, New York, NY, USA, 2004. ACM.

[37] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827, 2013.

[38] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.

[39] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *CoRR*, abs/1711.11586, 2017.