NAM: Non-Adversarial Unsupervised Domain Mapping

Yedid Hoshen[1] and Lior Wolf[1,2]

[1] Facebook AI Research
[2] Tel Aviv University

Abstract. Several methods were recently proposed for the task of translating images between domains without prior knowledge in the form of correspondences. The existing methods apply adversarial learning to ensure that the distribution of the mapped source domain is indistinguishable from the target domain, which suffers from known stability issues. In addition, most methods rely heavily on "cycle" relationships between the domains, which enforce a one-to-one mapping. In this work, we introduce an alternative method: Non-Adversarial Mapping (NAM), which separates the task of target domain generative modeling from the cross-domain mapping task. NAM relies on a pre-trained generative model of the target domain, and aligns each source image with an image synthesized from the target domain, while jointly optimizing the domain mapping function. It has several key advantages: higher quality and resolution image translations, simpler and more stable training and reusable target models. Extensive experiments are presented validating the advantages of our method.

1 Introduction

The human ability to think in spontaneous analogies motivates the field of unsupervised domain alignment, in which image to image translation is achieved without correspondences between samples in the training set. Unsupervised domain alignment methods typically operate by finding a function for mapping images between the domains so that after mapping, the distribution of mapped source images is identical to that of the target images.

Successful recent approaches, e.g. DTN [28], CycleGANs [37] and DiscoGAN [14], utilize Generative Adversarial Networks (GANs) [9] to model the distributions of the two domains, $\mathcal{X}$ and $\mathcal{Y}$. GANs are very effective tools for generative modeling of images, however they suffer from instability in training, making their use challenging. The instability typically requires careful choice of hyper-parameters and often multiple initializations due to mode collapse. Current methods also make additional assumptions that can be restrictive, e.g., DTN assumes that a pre-trained high-quality domain specific feature extractor exists which is effective for both domains. This assumption is good for the domain of faces (which is the main application of DTN) but may not be valid for all cases. CycleGAN and DiscoGAN make the assumption that a transformation

T_{XY} can be found for every $\mathcal{X}$-domain image x to a unique $\mathcal{Y}$-domain image y, and another transformation T_{YX} exists between the $\mathcal{Y}$ domain and the original X-domain image $y = T_{XY}(x)$, $x = T_{YX}(y)$. This is problematic if the actual mapping is many-to-one or one-to-many, as in super-resolution or coloring.

We propose a novel approach motivated by cross-domain matching. We separate the problem of modeling the distribution of the target domain from the source to target mapping problem. We assume that the target image domain distribution is parametrized using a generative model. This model can be trained using any state-of-the-art unconditional generation method such as GAN [25], GLO [2], VAE [15] or an existing graphical or simulation engine. Given the generative model, we solve an unsupervised matching problem between the input $\mathcal{Y}$ domain images and the $\mathcal{X}$ domain. For each source input image y, we synthesize an $\mathcal{X}$ domain image $G(z_y)$, and jointly learn the mapping function $T()$, which maps images from the X domain to the $\mathcal{Y}$ domain. The synthetic images and mapping function are trained using a reconstruction loss on the input $\mathcal{Y}$ domain images.

Our method is radically different from previous approaches and it presents the following advantages:

1. A generative model needs to be trained only once per target dataset, and can be used to map to this dataset from all source datasets without adversarial generative training.
2. Our method is one-way and does not assume a one-to-one relationship between the two domains, e.g., it does not use cycle-constraints.
3. Our work directly connects between the vast literature of unconditional image generation and the task of cross-domain translation. Any progress in unconditional generation architectures can be simply plugged in with minimal changes. Specifically, we can utilize recent very high-resolution generators to obtain high quality results.

2 Previous Work

Unsupervised domain alignment: Mapping across similar domains without supervision has been successfully achieved by classical methods such as Congealing [22]. Unsupervised translation across very different domains has only very recently began to generate strong result, due to the advent of generative adversarial networks (GANs), and all state-of-the-art unsupervised translation methods we are aware of employ GAN technology . As this constraint is insufficient for generating good translations, current methods are differentiated by additional constraints that they impose.

The most popular constraint is cycle-consistency: enforcing that a sample that is mapped from $\mathcal{X}$ to $\mathcal{Y}$ and back to $\mathcal{X}$, reconstructs the original sample. This is the approach taken by DiscoGAN [14], CycleGAN [37] and DualGAN [30]. Recently, StarGAN [5] created multiple cycles for mapping in any direction between multiple (two or more) domains. The generator receives as input the source image as well as the specification of the target domain.

For the case of linear mappings, orthogonality has a similar effect to circularity. Very recently, it was used outside computer vision by several methods [33, 34, 6, 12] for solving the task of mapping words between two languages without using parallel corpora.

Another type of constraint is provided by employing a shared latent space. Given samples from two domains $\mathcal{X}$ and $\mathcal{Y}$, CoGAN [21], learns a mapping from a random input vector z to matching samples, one in each domain. The domains $\mathcal{X}$ and $\mathcal{Y}$ are assumed to be similar and their generators (and GAN discriminators) share many of the layers' weights, similar to [27]. Specifically, the earlier generator layers are shared while the top layer are domain specific. CoGAN can be modified to perform domain translation in the following way: given a sample $x \in \mathcal{X}$, a latent vector z_x is fitted to minimize the distance between the image generated by the first generator $G_{\mathcal{X}}(z_x)$ and the input image x. Then, the analogous image in $\mathcal{Y}$ is given by $G_{\mathcal{Y}}(z_x)$. This method was shown in [37] to be less effective than cycle-consistency based methods.

UNIT [20] employs an encoder-decoder pair per each domain. The latent spaces of the two are assumed to be shared, and similarly to CoGAN, the layers that are distant from the image (the top layers of the encoder and the bottom layers of the encoder) are shared between the two domains. Cycle-consistency is added as well, and structure is added to the latent space using variational autoencoder [16] loss terms.

As mentioned above our method does not use adversarial or cycle-consistency constraints.

Mapping using Domain Specific Features Using domain specific features has been found by DTN [28] to be important for some tasks. It assumed that a feature extractor can be found, for which the source and target would give the same activation values. Specifically it uses face specific features to map faces to emojis. While for some of the tasks, our work does use a "perceptual loss" that employs a pretrained imagenet-trained network, this is a generic feature extraction method that is not domain specific. We claim therefore that our method still qualifies as unsupervised. For most of the tasks presented, the VGG loss alone, would not be sufficient to recover good mappings between the two domains, as shown in ANGAN [11].

Unconditional Generative Modeling: Many methods were proposed for generative models of image distributions. Currently the most popular approaches rely on GANs and VAEs [15]. GAN-based methods are plagued by instability during training. Many methods were proposed to address this issue for unconditional generation, e.g., [1, 10, 23]. The modifications are typically not employed in cross-domain mapping works. Our method trains a generative model (typically a GAN), in the $\mathcal{X}$ domain separately from any $\mathcal{Y}$ domain considerations, and can directly benefit from the latest advancements in the unconditional image generation literature. GLO [3] is an alternative to GAN, which iteratively fits per-image latent vectors (starting from random "noise") and learns a mapping $G()$ between the noise vectors and the training images. GLO is trained using a

reconstruction loss, minimizing the difference between the training images and those generated from the noise vectors. Differently from our approach is tackles unconditional generation rather than domain mapping.

3 Unsupervised Image Mapping without GANs

In this section, we present our method - NAM - for unsupervised domain mapping. The task we aim to solve, is finding analogous images across domains. Let $\mathcal{X}$ and $\mathcal{Y}$ be two image domains, each with some unique characteristics. For each domain we are given a set of example images. The objective is to find for every image y in the $\mathcal{Y}$ domain, an analogous image x which appears to come from the $\mathcal{X}$ domain but preserves the unique content of the original y image.

3.1 Non-Adversarial Exact Matching

To motivate our approach, we first consider the simpler case, where we have two image domains $\mathcal{X}$ and $\mathcal{Y}$, consisting of sets of images $\{x_i\}$ and $\{y_i\}$ respectively. We assume that the two sets are approximately related by a transformation T, and that a matching paired image x exists for every image y in domain $\mathcal{Y}$ such that $T(x) = y$. The task of matching becomes a combination of two tasks: i) inferring the transformation between the two domains ii) finding matching pairs across the two domains. Formally this becomes:

$$L = \sum_i \|T(\sum_j M_{ij} x_j), y_i\| \tag{1}$$

Where M_{ij} is the matching matrix containing $M_{i,j} = 1$ if x_j and y_i are matching and 0 otherwise. The optimization is over both the transformation $T()$ as well as binary match matrix M.

Since the optimization of this problem is hard, a relaxation method - *ANGAN* - was recently proposed [11]. The binary constraint on the matrix was replaced by the requirement that $M_{ij} \geq 0$ and $\sum_j M_{ij} = 1$. As optimization progresses, a barrier constraint on M, pushes the values of M to 0 or 1.

ANGAN was shown to be successful in cases where exact matches exist and $T()$ is initialized with a reasonably good solution obtained by CycleGAN.

3.2 Non-Adversarial Inexact Matching

In Sec. 3.1, we described the scenario in which exact matches exist between the images in domains $\mathcal{X}$ and $\mathcal{Y}$. In most situations, exact matches do not exist between the two domains. In such situations it is not sufficient to merely find an image x in the domain $\mathcal{X}$ training set such that for a target $\mathcal{Y}$ domain image y, we have $y = T(x)$ as we cannot hope that such a match will exist. Instead, we need to synthesize an image $\tilde{x}$ that comes from the $\mathcal{X}$ domain distribution,

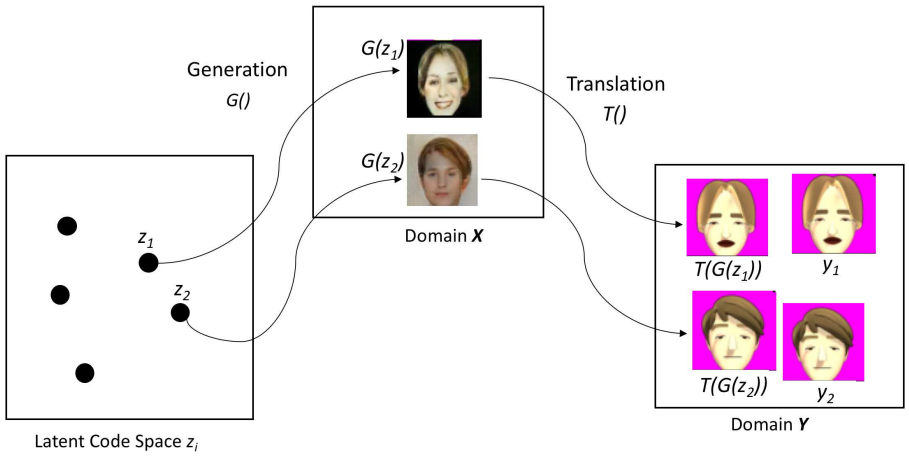

Fig. 1. Given a generator G for domain $\mathcal{X}$ and training samples $\{y_i\}$ in domain $\mathcal{Y}$, NAM jointly learns the transformation $T : \mathcal{X} \to \mathcal{Y}$ and the latent vectors $\{z_i\}$ that give rise to samples $\{T(G(z_i))\}$ that resemble the training images in $\mathcal{Y}$

and satisfies $y = T(\tilde{x})$. This can be achieved by removing the stochasticity requirement in Eq. 1. Effectively, this models the images in the $\mathcal{X}$ domain as:

$$\tilde{x} = \sum_j \alpha_i x_i \tag{2}$$

This solution is unsatisfactory on several counts: (i) the simplex model for the $\mathcal{X}$ domain cannot hope to achieve high quality image synthesis for general images (ii) The complexity scales quadratically with the number of training images making both training and evaluation very slow.

3.3 Non-Adversarial Mapping (NAM)

In this section we generalize the ideas presented in the previous sections into an effective method for mapping between domains without supervision or the use of adversarial methods.

In Sec. 3.2, we showed that to find analogies between domains $\mathcal{X}$ and $\mathcal{Y}$, the method requires two components: (i) a model for the distribution of the $\mathcal{X}$ domain, and (ii) a mapping function $T()$ between domains $\mathcal{X}$ and $\mathcal{Y}$.

Instead of the linear simplex model of Sec. 3.2, we propose to model the $\mathcal{X}$ domain distribution by a neural generative model $G(z)$, where z is a latent vector. The requirements on the generative model $G()$ are such that for every image x in the $\mathcal{X}$ domain distribution we can find z such $x = G(z)$ and that $G()$ is compact, that is, for no z, will $G(z)$ lie outside the $\mathcal{X}$ domain. The task of learning such generative models, is the research focus of several communities. In this work we do not aim to contribute to the methodology of unsupervised

generative modeling, but rather use the state-of-the-art modeling techniques obtained by previous approaches, for our generator $G()$. Methods which can be used to obtain generative model $G()$ include: GLO [2], VAE [15], GAN [9] or a hand designed simulator (see for example [29]). In our method, the task of single domain generative modeling is entirely decoupled from the task of cross-domain mapping, which is the one we set to solve.

Armed with a much better model for the $\mathcal{X}$ domain distribution, we can now make progress on finding synthetic analogies between $\mathcal{X}$ and $\mathcal{Y}$. Our task is to find for every $\mathcal{Y}$ domain image y, a synthetic $\mathcal{X}$ domain image $G(z_y)$ so that when mapped to the $\mathcal{Y}$ domain $y = T(G(z_y))$. The task is therefore twofold: (i) for each y, we need to find the latent vector z_y which will synthesize the analogous $\mathcal{X}$ domain image, and (ii) the mapping function $T()$ needs to be learned.

The model can therefore be formulated as an optimization problem, where the objective is to minimize the reconstruction cost of the training images of the $\mathcal{Y}$ domain. The optimization is over the latent codes, a unique latent code z_y vector for every input $\mathcal{Y}$ domain image y, as well as the mapping function $T()$. It is formally written as below:

$$argmin_{T,z_y} \sum_{y \in B} \|T(G(z_y)), y\| \tag{3}$$

The model is fully differentiable, as both the generative model $G()$ and the mapping function $T()$ are parameterized by neural networks. The above objective is jointly optimized for z_y and $T()$, but not for $G()$ which is kept fixed. The method is illustrated in Fig. 1.

3.4 Perceptual Loss

Although the optimization described in Sec. 3.3 can achieve good solutions, we found that introducing a perceptual loss, can significantly help further improve the quality of analogies. Let $\phi_i()$ be the features extracted from a deep-network at the end of the i'th block (we use VGG [26]). The perceptual loss is given by:

$$\|.,.\|_{VGG} = \sum_i \|\phi_i(T(G(z_y))), \phi_i(y)\|_1 + \|T(G(z_y)), y\|_1 \tag{4}$$

The final optimization problem becomes:

$$argmin_{T,z_y} \sum_{y \in B} \|T(G(z_y)), y\|_{VGG} \tag{5}$$

The VGG perceptual loss was found by several recent papers [4, 35] to give perceptually pleasing results. There have been informal claims in the past that methods using perceptual loss functions should count as supervised. We claim that the perceptual loss does not make our method supervised, as the VGG network does not come from our domains and does not require any new labeling effort. Our view is that taking advantage of modern feature extractors will benefit the field of unsupervised learning in general and unsupervised analogies in particular.

3.5 Inference and Multiple Solutions

Once training has completed, we are now in possession of the mapping function $T()$ which is now fixed (the pre-trained $G()$ was never modified as a part of training).

To infer the analogy of a new $\mathcal{Y}$ domain image y, we need to recover the latent code z_y which would yield the optimal reconstruction. The mapping function $T()$ is now fixed, and is not modified after training. Inference is therefore performed via the following optimization:

$$argmin_{z_y} \|T(G(z_y)), y\| \tag{6}$$

The synthetic $\mathcal{X}$ domain image $G(z_y)$ is our proposed solution to $\mathcal{Y}$ domain image y.

This inference procedure is a non-convex optimization problem. Different initializations, yield different final analogies. Let us denote initialization z_0^t where t is the ID of the solution. At the end of the optimization procedure for each initialization, the synthetic images $G(z^t)$ yield multiple proposed analogies for the task. We find $G(z^0)...G(z^T)$ are very diverse when in fact many analogies are available. For example, when the $\mathcal{X}$ domain is Shoes and the $\mathcal{Y}$ domain is Edges, there are many shoes that can result in the same edge image.

3.6 Implementation Details

In this section we give a detailed description of the procedure used to generate the experiments presented in this paper.

$\mathcal{X}$ domain generative model $G(.)$: Our method takes as input a pre-trained generative model for the $\mathcal{X}$ domain. In our MNIST, SVHN and cars, Edges2 (Shoes,Handbags) experiments, we used DCGAN [25] with (32,32,32,100,100) latent dimensions. The low resolution face image generator was trained on celebA and used the training method of [23]. The high resolution face generator is provided by [13] and the Dog generator by [32]. The hyperparameters of all trained generators were set to their default value. In our experiments GAN unconditional generators provided more compelling results than competing SOTA methods such as GLO and VAE.

Mapping function $T(.)$: The mapping function was designed so that it is powerful enough but not too large as to overfit. Additionally, it needs to preserve image locality, in the case of spatially aligned domains. We elected to use a network with an architecture based on [4]. We found that as we only rely on the networks to find correspondences rather than generate high-fidelity visual outputs, small networks were the preferred choice. We used a similar architecture to [4], with a single layer per scale, and linearly decaying number of filters per layer starting with $4F$, and decreasing by F with every layer. $F = 8$ for SVHN and MNIST and $F = 32$ for the other experiments.

Optimization: We optimized using SGD with ADAM [17]. For all datasets we used a learning rate of 0.03 for the latent codes z_y and 0.001 for the mapping

Fig. 2. Converting digits between SVHN and MNIST (both directions). (a) CycleGAN results (b) NAM results (c) the input images.

function $T(.)$ (due to the uneven update rates of each z_y and $T()$). On all datasets training was performed on 2000 randomly selected examples (a subset) from the $\mathcal{Y}$ domain. Larger training sets were not more helpful as each z_y is updated less frequently.

Generating results: The $\mathcal{X}$ domain translation of $\mathcal{Y}$ domain image y is given by $G(z_y)$, where z_y is the latent code found in optimization. The $\mathcal{X} \to \mathcal{Y}$ mapping $T(x)$, typically resulted in weaker results due to the relatively shallow architecture selected for $T(.)$. A strong $T(.)$ can be trained by calculating a set of $G(z_y)$ and y (obtained using NAM), and training a fully-supervised network $T(.)$, e.g. as described by [4]. A similar procedure was carried out in [11].

4 Experiments

To evaluate the merits of our method, we carried out an extensive set of qualitative and quantitative experiments.

SVHN-MNIST Translation: We evaluated our method on the SVHN-MNIST translation task. Although SVHN [24] and MNIST [18] are simple datasets, the mapping task is not trivial. The MNIST dataset consists of simple handwritten single digits written on black background. In contrast, SVHN images are taken from house numbers and typically contain not only the digit of interest but also parts of the adjacent digits, which are nuisance information. We translate in both directions SVHN→MNIST and MNIST→SVHN. The results are presented in Fig. 2. We can observe that in the easier direction of SVHN→MNIST, in which there is information loss, NAM resulted in more accurate translations than CycleGAN. In the reverse direction of MNIST→SVHN, which is harder due to information gain, CycleGAN did much worse, whereas NAM was often successful. Note that perceptual loss was not used in the MNIST→SVHN translation task.

We performed a quantitative evaluation of the quality of SVHN↔MNIST translation. This was achieved by mapping an image from the one dataset to appear like the other dataset, and classifying it using a pre-trained classifier trained on the clean target data (the classifier followed a NIN architecture [19],

Table 1. Translation quality measured by translated digit classification accuracy (%)

	SVHN→MNIST	*MNIST→SVHN*
CycleGAN	26.8	17.7
NAM	33.3	31.9

and achieved test accuracies of around 99.5% on MNIST and 95.0% on SVHN). The results are presented in Tab. 1. We can see that the superior translations of NAM are manifested in higher classification accuracies.

Edges2Shoes: The task of mapping edges to shoes is commonly used to qualitatively evaluate unsupervised domain mapping methods. The two domains are a set of Shoe images first collected by [31], and their edge maps. The transformation between an edge map and the original photo-realistic shoe image is non-trivial, as much information needs to be hallucinated.

Fig. 3. (a) Comparison of NAM and DiscoGAN for Edges2Shoes. Each triplet shows NAM (center row) vs. DiscoGAN (top row) for a given input (bottom row). (b) A similar visualization for Edges2Handbags. (c,d) NAM mapping from a single source edge image (shown first) for different random initializations.

Examples of NAM and DiscoGAN results can be seen in Fig. 3(a). The higher quality of the analogies generated by NAM is apparent. This stems from using a pre-learned generative model rather than learning jointly with mapping, which is hard and results in worse performance. We also see the translations result in more faithful analogies. Another advantage of our method is the ability to map one input into many proposed solutions. Two examples are shown in Fig. 3(c) and (d). It is apparent that the solutions all give correct analogies, however they give different possibilities for the correct analogy. This captures the one-to-many property of the edge to shoes transformation.

Fig. 4. Comparison of NAM results for different generators

Fig. 5. Example results for mapping from bags (original images - top) to shoes. NAM mapped images (center) are clearly better than DiscoGAN mapped images (bottom).

As mentioned in the method description, NAM requires high-quality generators, and performs better for better pre-trained generators. In Fig. 4 we show NAM results for generators trained with: VAE [15] with high (VAE-h) and low (VAE-l) regularization, GLO [2], DCGAN [25] and Spectral-Normalization GAN [23]. We can see from the results that NAM works is all cases. however results are much better for the best generators (DCGAN, Spectral-Norm GAN).

Edges2Handbags: The Edges2Handbags [36] dataset is constructed similarly to Edges2Shoes. Sample results on this dataset can be seen in Fig. 3(b). The conclusions are similar to Edges2Shoes: NAM generates analogies that are both more appealing and more precise than DiscoGAN.

Shoes2Handbags: One of the major capabilities displayed by DiscoGAN is being able to relate domains that are very different. The example shown in [14], of mapping images of handbags to images of shoes that are semantically related, illustrates the ability of making distant analogies.

In this experiment we show that NAM is able to make analogies between handbags and shoes, resulting in higher quality solutions than those obtained by DiscoGAN. In order to achieve this, we replace the reconstruction VGG loss by

Table 2. Car2Car root median residual deviation from linear alignment (lower is better).

DiscoGAN	*NAM*
13.81	1.47

a Gram matrix VGG loss, as used in Style Transfer [8]. DiscoGAN also uses a Gram matrix loss (with feature extracted from its discriminator). For this task, we also add a skip connection from $G(z)$, as the domains are already similar under a style loss.

Example images can be seen in Fig. 5. The superior quality of the NAM generated mapped images is apparent. The better quality is a result of using an interpretable and well understood non-adversarial loss which is quite straight forward to optimize. Another advantage comes from being able to "plug-in" a high-quality generative model.

Car2Car: The Car2Car dataset is a standard numerical baseline for cross-domain image mapping. Each domain consists of a set of different cars, presented in angles varying from -75 tp 75 degrees. The objective is to align the two domains such that a simple relationship exists between orientation of car image y and mapped image x (typically, either the orientation of x and y should be equal or reversed). A few cars mapped by NAM and DiscoGAN can be seen in Fig. 6. Our method results in a much cleaner mapping. We also quantitatively evaluate the mapping, by training a simple regressor on the car orientation in the $\mathcal{X}$ domain, and comparing the ground-truth orientation of y with the predicted orientation of the mapped image x. We evaluate using the root median residuals (as the regressor sometimes flips orientations of -75 to 75 resulting in anomalies). For car2car, we used a skip connection from $G(z)$ to the output. Results are seen in Tab. 2. Our method significantly outperforms DiscoGAN. Interestingly, on this task, on this task, it was not necessary to use a perceptual loss, a simple Euclidean pixel loss was sufficient for a very high-quality solution on this task. As a negative result, on the car2head task i.e. mapping between car images and images of heads of different people at different azimuth angles; NAM did not generate a simple relation between the orientations of the cars and heads but a more complex relationship. Our interpretation from looking at results is that black cars were inversely correlated with the head orientation, whereas white cars were positively correlated.

Avatar2Face: One of the first applications of cross-domain translation was face to avatar generation by DTN [28]. This was achieved by using state-of-the-art face features, and ensured the features are preserved in the original face and the output avatar (f-constancy). Famously however, DTN does not generate good results on avatar2face generation, which involves adding rather than taking away information. Due to the many-to-one nature of our approach, NAM is better suited for this task. In fig. 7 we present example images of our avatar2face conversions. This was generated by a small generative model with a DCGAN [25]

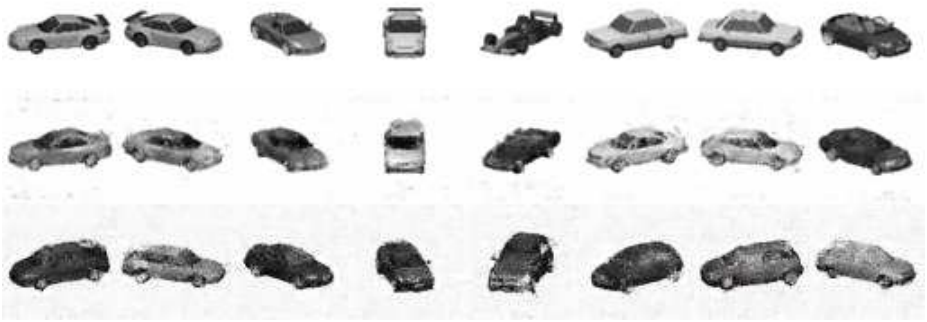

Fig. 6. Example results for mapping across two sets of car models at different orientations. Although DiscoGAN (bottom) does indeed preserve orientation of the original images (top) to some extent, NAM (center) preserves both orientation and general car properties very accurately - despite the target domain containing few sports cars.

Fig. 7. Example results for mapping Avatars (top) to Faces (bottom) using NAM.

architecture, trained using Spectral Normalization GAN [23] using celebA face images. The avatar dataset was obtained from the authors of [29].

Plugging in State-of-the-Art Generative models: One of the advantages of our method is the independence between mapping and generative modeling. The practical consequence is that any generative model, even very large models that take weeks to train, can be effortlessly plugged into our framework. We can then map any suitable source domain to it, very quickly and efficiently.

Amazing recent progress has been recently carried out on generative modeling. One of the most striking examples of it is Progressive Growing of GANs (PGGAN) [13], which has yielded generative models of faces with unprecedented resolutions of 1024X1024. The generative model training took 4 days of 8 GPUs, and the architecture selection is highly non-trivial. Including the training of such generative models in unsupervised domain mapping networks is therefore very hard.

For NAM, however, we simply set $G()$ as the trained generative model from the authors' code release. A spatial transformer layer, with parameters optimized by SGD per-image, reduced the model outputs to the Avatar scale (which we

Fig. 8. One-to-many high-resolution mapping from Avatars to Faces using the pre-trained generator from [13]

chose to be $64X64$). We present visual results in Fig. 9. Our method is able to find very compelling analogous high-resolution faces. Scaling up to such high resolution would be highly-nontrivial with state-of-the-art domain translation methods. We mention that DTN [28], the state-of-the-art approach for unsupervised face-to-emoji mapping, has not been successful at this task, even though it uses domain specific facial features.

To show the generality of our approach, we also mapped Avatars to Dog images. The generator was trained using StackGAN-v2 [32]. We plugged in the trained generators from the publicly released code into NAM. Although emoji to dogs is significantly more distant than emoji to human face (all the Avatars used, were human faces), NAM was still able to find compelling analogies.

5 Discussion

Human knowledge acquisition typically combines existing knowledge with new knowledge obtained from novel domains. This process is called blending [7]. Our work (as most of the existing literature) focuses on the mapping process i.e. being able to relate the information from both domains, but does not deal with the actual blending of knowledge. We believe that blending, i.e., borrowing from

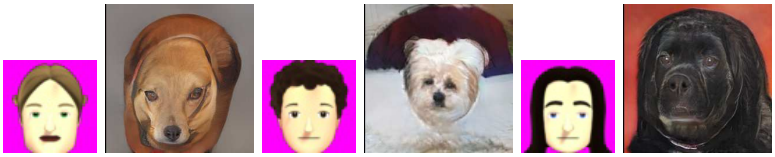

Fig. 9. High-resolution mapping from Avatars to Dogs, using the pre-trained generator from [32].

both domains to create a unified view that is richer than both sources would be an extremely potent direction for future research.

An attractive property of our model, is the separation between the acquisition of the existing knowledge and the fitting of a new domain. The preexisting knowledge is modeled as the generative model of domain $\mathcal{X}$, given by G; The fitting process includes the optimization of a learned mapper from domain $\mathcal{X}$ to domain $\mathcal{Y}$, as well as identifying exemplar analogies $G(z_y)$ and y.

A peculiar feature of our architecture, is that function $T()$ maps from the target ($\mathcal{X}$ domain) to the source ($\mathcal{Y}$ domain) and not the other way around. Mapping in the other direction would fail, since it can lead to a form of mode-collapse, in which all $\mathcal{Y}$ samples are mapped to the same generated $G(z)$ for a fixed z. While additional loss terms and other techniques can be added in order to avoid this, mode collapse is a challenge in generative systems and it is better to avoid the possibility of it altogether. Mapping as we do avoids this issue.

6 Conclusions

Unsupervised mapping between domains is an exciting technology with many applications. While existing work is currently dominated by adversarial training, and relies on cycle constraints, we present results that support other forms of training.

Since our method is very different from the existing methods in the literature, we have been able to achieve success on tasks that do not fit well into other models. Particularly, we have been able to map low resolution face avatar images into very high resolution images. On lower resolution benchmarks, we have been able to achieve more visually appealing and quantitatively accurate analogies.

Our method relies on having a high quality pre-trained unsupervised generative model for the $\mathcal{X}$ domain. We have shown that we can take advantage of very high resolution generative models, e.g., [13, 32]. As the field of unconditional generative modeling progresses, so will the quality and scope of NAM.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)

2. Bojanowski, P., Joulin, A., Lopez-Paz, D., Szlam, A.: Optimizing the latent space of generative networks. arXiv preprint arXiv:1707.05776 (2017)
3. Bojanowski, P., Joulin, A., Lopez-Paz, D., Szlam, A.: Optimizing the latent space of generative networks. arXiv preprint arXiv:1707.05776 (2017)
4. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. ICCV (2017)
5. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. arXiv preprint arXiv:1711.09020 (2017)
6. Conneau, A., Lample, G., Ranzato, M., Denoyer, L., Jégou, H.: Word translation without parallel data. arXiv preprint arXiv:1710.04087 (2017)
7. Fauconnier, G., Turner, M.: The Way We Think: Conceptual Blending and the Mind's Hidden Complexities. Basic Books (2002)
8. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: CVPR (2016)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS, pp. 2672–2680 (2014)
10. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems. pp. 5769–5779 (2017)
11. Hoshen, Y., Wolf, L.: Identifying analogies across domains. International Conference on Learning Representations (2018)
12. Hoshen, Y., Wolf, L.: An iterative closest point method for unsupervised word translation. arXiv preprint arXiv:1801.06126 (2018)
13. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
14. Kim, T., Cha, M., Kim, H., Lee, J., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. arXiv preprint arXiv:1703.05192 (2017)
15. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
16. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. stat **1050**, 10 (2014)
17. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: The International Conference on Learning Representations (ICLR) (2016)
18. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010)
19. Lin, M., Chen, Q., Yan, S.: Network In Network. In: ICLR (2014)
20. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Advances in Neural Information Processing Systems. pp. 700–708 (2017)
21. Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. In: NIPS, pp. 469–477 (2016)
22. Miller, E.G., Matsakis, N.E., Viola, P.A.: Learning from one example through shared densities on transforms. In: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on. vol. 1, pp. 464–471. IEEE (2000)
23. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (2018)
24. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2011)

25. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ICLR (2015)
27. Sutskever, I., Jozefowicz, R., Gregor, K., Rezende, D., Lillicrap, T., Vinyals, O.: Towards principled unsupervised learning. In: ICLR workshop (2016)
28. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. In: International Conference on Learning Representations (ICLR) (2017)
29. Wolf, L., Taigman, Y., Polyak, A.: Unsupervised creation of parameterized avatars. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1530–1538 (2017)
30. Yi, Z., Zhang, H., Tan, P., Gong, M.: Dualgan: Unsupervised dual learning for image-to-image translation. arXiv preprint arXiv:1704.02510 (2017)
31. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: CVPR (2014)
32. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.: Stackgan++: Realistic image synthesis with stacked generative adversarial networks. arXiv: 1710.10916 (2017)
33. Zhang, M., Liu, Y., Luan, H., Sun, M.: Adversarial training for unsupervised bilingual lexicon induction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 1959–1970 (2017)
34. Zhang, M., Liu, Y., Luan, H., Sun, M.: Earth mover's distance minimization for unsupervised bilingual lexicon induction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 1934–1945 (2017)
35. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. arXiv preprint arXiv:1801.03924 (2018)
36. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: European Conference on Computer Vision. pp. 597–613. Springer (2016)
37. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networkss. arXiv preprint arXiv:1703.10593 (2017)