Image Disentanglement and Uncooperative Re-Entanglement for High-Fidelity Image-to-Image Translation

Adam W. Harley
Carnegie Mellon University
aharley@cmu.edu

Shih-En Wei
Facebook Reality Labs
shih-en.wei@oculus.com

Jason Saragih
Facebook Reality Labs
jason.saragih@oculus.com

Katerina Fragkiadaki
Carnegie Mellon University
katef@cs.cmu.edu

Abstract

Cross-domain image-to-image translation should satisfy two requirements: (1) preserve the information that is common to both domains, and (2) generate convincing images covering variations that appear in the target domain. This is challenging, especially when there are no example translations available as supervision. Adversarial cycle consistency was recently proposed as a solution [29], with beautiful and creative results, yielding much follow-up work. However, augmented reality applications cannot readily use such techniques to provide users with compelling translations of real scenes, because the translations do not have high-fidelity constraints. In other words, current models are liable to change details that should be preserved: while re-texturing a face, they may alter the face's expression in an unpredictable way. In this paper, we introduce the problem of high-fidelity image-to-image translation, and present a method for solving it. Our main insight is that low-fidelity translations typically escape a cycle-consistency penalty, because the back-translator learns to compensate for the forward-translator's errors. We therefore introduce an optimization technique that prevents the networks from cooperating: simply train each network only when its input data is real. Prior works, in comparison, train each network with a mix of real and generated data. Experimental results show that our method accurately disentangles the factors that separate the domains, and converges to semantics-preserving translations that prior methods miss.

1. Introduction

Unpaired cross-domain image-to-image translation is achieving exceptionally convincing results in a variety of domains [29]. *High-fidelity* image translation requires not only credible translations, but also strict preservation of the factors that are common to both domains. Consider Fig-
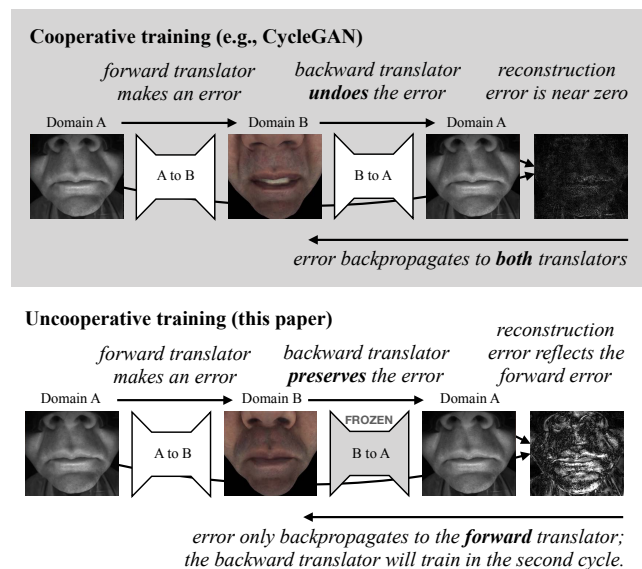

Figure 1: "Coooperative" training setup from prior work (top), compared with our proposed "uncooperative" training (bottom). Preventing cooperation between the forward and back-translation yields a more useful reconstruction loss.

ure 1. We wish to translate an image of a face across two domains that mostly differ in texture. It is inappropriate for the translator to additionally change the face's expression. Unfortunately, this failure mode is surprisingly common in standard unsupervised image-to-image models.

Why does this happen? In the standard approach (based on CycleGAN [29]), there are two main networks: a forward translator, and a backward translator. There are also two main losses: an adversarial loss, which encourages the translated images to be indistinguishable from ones in their target domain, and a cycle consistency loss, which encourages that forward translation (*i.e.*, A to B) followed by backward translation (*i.e.*, B back to A) yields the original input
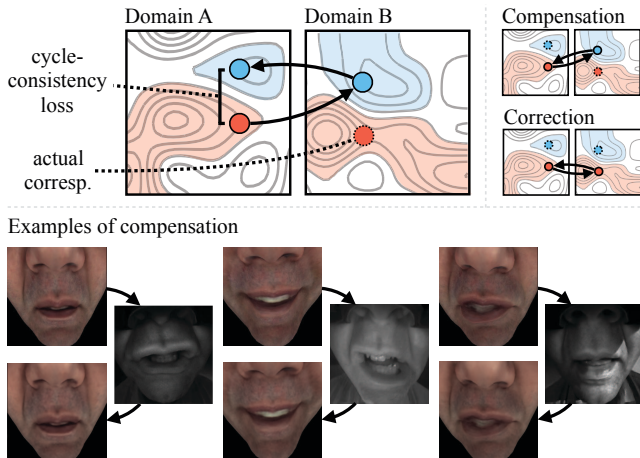
Figure 2: Top: Two strategies for optimizing the cycle-consistency loss: compensation and correction. Optimizing the cycle-consistency loss encourages the forward-translator to find better mappings (i.e., correction), but also encourages the back-translator to compensate for bad mappings (i.e., compensation). Bottom: real examples of compensation learned by a CycleGAN; cycle-consistency is being met, but the translations are not faithful.

(*i.e.*, forming a cycle). The problem with this setup is that there is no "fidelity loss" on the translation. In other words, the forward translator may generate arbitrary samples in domain B, and as long as the backward translator reconstructs the input, there is no penalty. One can add geometric constraints to the translation [6, 23], but these only approximate a loss on fidelity. We present an approach that directly penalizes unfaithful translations, by ensuring that forward-translation errors are *preserved* during back-translation, as shown in Figure 1 (bottom).

Our main insight is that cycles are problematic when the two mapping functions are allowed to *cooperate*. By "cooperate" we mean that they optimize for each other's outputs. CycleGAN and its many variants [29, 10, 16, 7, 18, 1] all have a cooperative training setup: in each cycle, the first translator receives a real input, and the second translator receives a fake input (*i.e.*, an attempted translation/disentanglement) which it back-translates, and *both* networks get penalized according to the reconstruction error. This essentially asks the second network to compensate for the first network's errors. This is counterproductive, because if the second network succeeds, then the first network need not improve. Given sufficient optimization time, these cooperative setups find extremely effective "cheats", in which subtle signals are encoded into low-fidelity forward translations and subsequently decoded to achieve near-perfect back-translation, thus defeating the reconstruction error [4]. This is illustrated in Figure 2.

Our main contribution is in preventing the networks from compensating for each other's errors, via a simple optimiza-

tion technique: simply train each network *only* when its input data is real. With this technique, neither network learns about the other's behavior, which renders cooperation impossible. Instead, the back-translator simply *preserves* any errors made during forward-translation, and the reconstruction penalty is put entirely on the forward translator. This forces the networks to learn more faithful mappings to their target domains.

Our "uncooperative" training also provides a route to unsupervised factor disentanglement. Several prior works have modified CycleGANs to perform a disentanglement and subsequent re-entanglement: the first translator disentangles the input into (1) an image in the second domain, and (2) a residual; the second translator entangles these to reconstruct the input [30, 10, 1]. In practice, however, an unconstrained "residual" path can actually be detrimental to the final results, since the model may exploit this path to encode the *entire input,* greatly facilitating the the cycle-consistency objective. Prior works have proposed a variety of methods to mitigate this problem, but usually at the cost of severely reducing the representational capacity of the residual (*e.g.*, limiting it to 8 dimensions), and making strong assumptions about its distribution (*e.g.*, assuming it is standard normal) [1, 10, 16, 18]. After applying these heavy constraints, some prior works report that the residual path simply goes *ignored* by the model, unless its usage is facilitated by careful design choices (*e.g.*, rather than simply concatenating the residual as an input, enforce its usage as layer-wise normalization coefficients, applied throughout the second translator) [1, 18]. Our optimization technique allows us to disentangle multi-scale high-dimensional residuals, without requiring parameter-sensitive representational constraints.

In experiments with real images, we show that our optimization method delivers an obvious qualitative improvement over the current state-of-the-art, both in terms of semantics-preservation and residual-factor disentanglement. In synthetic data (where the residual is known), we demonstrate that our "uncooperative" optimization leads to quantitatively accurate disentanglement, whereas "cooperative" optimization does not.

2. Related Work

Image-to-image translation has recently attracted great attention, partly thanks to the success of generative adversarial networks (GANs) [8, 21, 28, 14]. The goal in image-to-image translation is to translate an image in one domain to a corresponding image in the second domain. Pix2Pix [11] trains models for this task using paired data from the two domains (*i.e.*, input-output pairs, exemplifying good translations). CycleGAN [29] removes the need for paired data by forming a translation "cycle"—forward translation followed by backward translation—which per-

mits a natural reconstruction objective between the input and the back-translation. This is important, because in many domains, paired examples do not exist (*e.g.*, a face in the exact same pose/expression in two different physical environments). CycleGAN often preserves the structural content of the images, but this may simply be a consequence of the convolutional architecture [17]. CycleGAN is only capable of learning one-to-one mappings, but several works (not all unsupervised) have proposed variants that are capable of one-to-many mappings, such as Augmented CycleGAN [1], DRIT [16], MUNIT [10], BicycleGAN [30], and cross-domain disentanglers [7]. These methods are able to generate diverse image with similar "content" (*i.e.*, structural pattern) but different "style" (*i.e.*, textural rendering) through disentanglement. These methods use strong assumptions or regularizations to avoid undesirable local optima, including shared latent spaces [10, 16, 7], loss on KL divergence from simple Gaussians [10, 16, 30], or low-dimensional representations [16, 1, 30, 7]. The effectiveness of these methods is therefore highly dependent on parameter selection.

Image factor disentanglement is necessary if we wish to control the latent factors in the generated images. Hadad *et al.* [9] assumes the availability of attribute labels in a particular domain, where the goal is to disentangle images into a target domain plus a residual (*i.e.*, "everything else"). Many disentanglement works also make strong assumptions on domain knowledge of the latent space, which includes having data pre-grouped according to individual factors [24, 15], or having exact knowledge of the structure and function of individual factors (*e.g.*, for faces: identity, pose, shape, texture [26, 25]). In this work, we do not have attribute labels, we do not make assumptions on the latent space, and we perform disentanglement using only the unpaired image data. Similar to our method, InfoGAN [3] and MINE [2] are completely unsupervised, but the approach in these works is quite different: these methods maximize the mutual information between the inferred latent variables and the observations, while we use discriminators and reconstruction to achieve disentanglement.

3. Method

There are three key ingredients to our method: (1) adversarial priors, which encourages the translated images to be indistinguishable from ones in their target domain, (2) cycle-consistency, which encourages the translations to be invertible, and (3) "uncooperative" optimization, which ensures the networks do not "cheat" toward an undesirable local minimum.

3.1. Preliminaries

Let V and C be two image domains, such that the images $v \in V$ have more information than the images $c \in C$.

That is, V contains variation in some latent factor that is either constant or absent in C. This implies that the mapping $V \mapsto C$ is many-to-one, and the mapping $C \mapsto V$ is one-to-many. As a mnemonic, note that V is **v**ariable in some aspect where C is **c**onstant.

Let R be the **r**esidual information that is in V but not in C. Accessing this extra information allows us to form bijective (one-to-one) mappings, $V \mapsto C \times R$ and $C \times R \mapsto V$. Note that R is not necessarily an image domain. In our implementation, each $r \in R$ is a collection of deep featuremaps at multiple scales, which allows its actual form to be determined entirely by the data.

Our goal is to learn functions that can map between these domains. We call the first mapping a disentanglement, denoted **D**, since it performs an intricate splitting operation: $\mathbf{D}(v) = (c, r)$. We call the second mapping an entanglement, denoted **E**, since it performs a merging operation: $\mathbf{E}(c, r) = v$. Figure 3 relates the notation to the data and architecture. Note that **D** and **E** are inverses of one another.

Our input is a set of samples $\{c_i\}_{i=1}^{N}$ from C, and samples $\{v_j\}_{j=1}^{M}$ from V. The two datasets are unpaired, and true correspondences might not exist.

3.2. Adversarial priors

Our model has two main networks, **D** and **E**. We would like to have $\mathbf{D} : V \mapsto C \times R$, and $\mathbf{E} : C \times R \mapsto V$. To achieve this, we introduce adversarial networks $\mathbf{A}_C$ and $\mathbf{A}_V$, which learn and impose priors on the distributions of our networks' outputs.

The adversarial networks attempt to discriminate between real and fake (*i.e.*, generated) samples of the domains C and V. In our notation, we distinguish "fake" samples with a prime symbol. We train our main networks against the adversarial labels with the least-squares loss [20]:

$$\mathcal{L}_{\text{GAN}} = [\mathbf{A}_V(v') - 1]^2 + [\mathbf{A}_C(c') - 1]^2. \quad (1)$$

In a separate (but concurrent) optimization, we also train the parameters of the adversaries, with the losses $\mathcal{L}_{\mathbf{A}_C} = [\mathbf{A}_C(c) - 1]^2 + [\mathbf{A}_C(c')]^2$, and $\mathcal{L}_{\mathbf{A}_V} = [\mathbf{A}_V(v) - 1]^2 + [\mathbf{A}_V(v')]^2$.

Note that we have no priors on the r' samples generated by **D**, because there is no dataset of "true" r samples. Prior works manufactured a prior by assuming that R is a low-dimensional Gaussian distribution (*e.g.*, 8 dimensions, with zero mean and unit variance) [10, 16, 7, 18]. Here, we avoid this limiting assumption. We are able to do this because of our unique optimization procedure, detailed in Sec 3.4. However, we do obtain some constraints on R by enforcing cycle consistency, described next.

3.3. Cycle consistency

On each training step, the model runs two "cycles". Each cycle generates a reconstruction loss, which constrains the
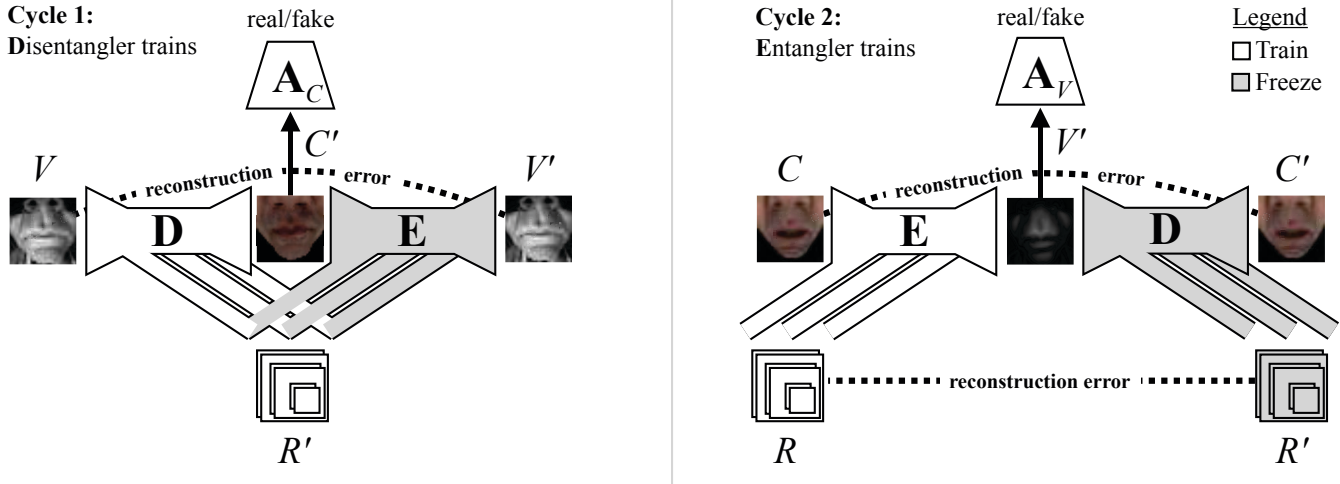
Figure 3: Disentanglement-Entanglement architecture. The networks are **D** (disentangler), **E** (entangler), $\mathbf{A}_C$ (adversary on C), and $\mathbf{A}_V$ (adversary on V). Primes indicate tensors that are generated by the model during the forward pass. Each generated tensor is subject to either reconstruction loss or an adversarial loss. R is a multi-scale output of **D**; it is concatenated to the featuremaps of **E** at the corresponding scales.

model to perform consistent forward-backward translation. Figure 3 illustrates the cycles.

In the first cycle, the disentangler **D** receives a random v from the dataset, and generates two outputs: $(c', r') = \mathbf{D}(v)$. These outputs are passed to the entangler **E**, which generates $v' = \mathbf{E}(c', r')$. If the disentanglement and re-entanglement are successful, this output should correspond to the original v. Therefore, we form the reconstruction objective $\ell_v = \|v - v'\|_1$, where $\|\cdot\|_1$ denotes the L1 norm. In summary, this cycle performs $\mathbf{E}(\mathbf{D}(v)) = v' \approx v$.

The second cycle is symmetric to the first. The entangler **E** receives a random c from the dataset, and an r generated from a random v. Note that it is necessary to use generated r samples here, since R is completely determined by the network. We omit the prime on this r since it is treated as an input rather than an output. From the input (c, r), the entangler generates $v' = \mathbf{E}(c, r)$. We then pass v' to the disentangler, which generates two new outputs $(c', r') = \mathbf{D}(v')$. If the entanglement and disentanglement are successful, these outputs should correspond to the original inputs. We therefore form the reconstruction objectives $\ell_c = \|c - c'\|_1$ and $\ell_r = \|r - r'\|_1$. In summary, this cycle performs $\mathbf{D}(\mathbf{E}(c, r)) = (c', r') \approx (c, r)$.

Collecting the reconstruction objectives, we have

$$\mathcal{L}_{\text{recon}} = \lambda_v \ell_v + \lambda_c \ell_c + \lambda_r \ell_r. \qquad (2)$$

Observe that there is no "fidelity" objective on the translated tensor of each cycle (*i.e.*, c' in Cycle 1, and v' in Cycle 2); these tensors only have an adversarial loss. In other words, there is nothing in the design to force c' to correspond to v, or v' to correspond to c, other than the

back-translation error. As we will show in experiments, this back-translation requirement is not sufficient, because the networks are able to cooperate on the back-translation: when **E** is the back-translator, it can compensate for errors made by **D**, and vice versa.

In practice, many of these "errors" are never corrected. Instead, they are adapted and refined, to minimize the adversarial loss while facilitating reconstruction. We call these "cheats": undesirable outputs that yield near-zero loss. At convergence, cheats often take the form of a within-domain transformation: this causes the adversary to not impose a loss (since the output is still in the correct domain), yet allows the second network to (jointly) learn how to undo the transformation. These cheats are especially visible in experiments with faces, likely because humans are so sensitive to faces [5]. Figures 1 and 5 show clear examples of this cheating behavior: while the two domains only differ in texture/lighting, the networks learn to additionally (and unpredictably) alter the facial expression.

We observe that this undesirable solution to the reconstruction error requires both **D** and **E** to be complicit in the scheme. For example, if **D** transforms its input while translating it, but **E** is unaware of the cheat, **E** will not undo the transformation while back-translating, yielding a loss. This leads us to our optimization procedure, which essentially prevents **D** and **E** from cooperating in this way.

3.4. Uncooperative optimization

The total loss we wish to minimize is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{GAN}}$. As long as the forward translations land in the target domains, $\mathcal{L}_{\text{GAN}}$ is minimized; as long as the backward translations reconstruct the inputs,

$\mathcal{L}_{\text{recon}}$ is minimized.

As explained, there is a local minimum to this loss, in which forward translation includes an undesirable transformation, and back-translation includes an inverse transformation. This ruins the fidelity of the translation.

To reach this bad minimum, each network needs to learn *two* functions: (1) translating "real" inputs into outputs in a target domain, and (2) decoding "fake" inputs by undoing generated translations. Referring to Figure 3, the disentangler **D** learns the first task in Cycle 1, and learns the second task in Cycle 2; the entangler **E** learns these functions in the opposite order. To prevent this from happening, we prevent the networks from learning how to decode fake inputs. We do this by *freezing* the networks when they receive fake inputs. When a network is "frozen", it is treated as a fixed but differentiable function, so that gradients flow through it, but it does not learn. Referring again to Figure 3, this means training **D** only in the first cycle, and training **E** only in the second cycle (where they respectively receive real inputs).

With this optimization technique, the networks are incapable of learning how to compensate for each other's errors. This means that an erroneous forward translation will always be taken "at face value" by the backward translator, and produce an appropriate loss. This is because the backward translator's only experience (in terms of gradient steps) comes from real data.

This method is a type of alternating optimization, in the sense that we keep one set of parameters fixed while optimizing the other set, and alternate. In practice, we alternate on every step. Specifically, we do a forward pass through Cycle 1, freeze **E** while we update **D**, and then do a forward pass through Cycle 2, and freeze **D** while we update **E**.

Including the independent update required for the adversarial networks, this setup requires three optimizers in total.

3.5. Implementation details

Network architecture Our implementation is based on CycleGAN [29]. The translators' architecture originally comes from Johnson *et al.* [12]: two stride-2 convolutions, four residual blocks, and two transposed convolutions.

We implement the disentangler as two separate networks: one for the C stream and one for the R stream; the R stream ends before the transposed convolutions. We found this worked significantly better than using a single network to produce both C and R.

The entangler uses the same architecture, except it receives skip connections from the R stream. There are three such connections: the first uses the featuremap produced after the stride-2 convolutions; the second uses the featuremap after two residual blocks, and the third uses the featuremap after the next (and final) two residual blocks. These R featuremaps are simply concatenated with the corresponding featuremaps in **E**. The intent with multiple skip connec-
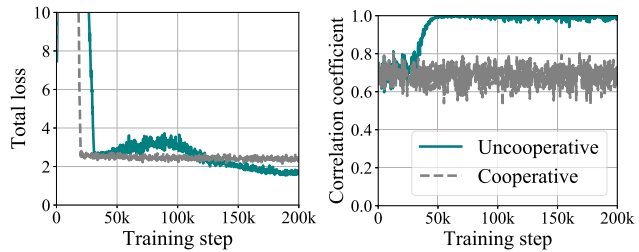

Figure 4: Uncooperative *vs.* cooperative optimization results on the objective function (left) and correlation with the ground-truth latent factor (right), over training steps. At convergence, uncooperative optimization achieves near-perfect disentanglement of the latent factor, whereas cooperative optimization does not.

tions is to allow the network the capacity to transfer residuals at multiple levels of scale and abstraction. Our model has fewer residual blocks than CycleGAN, but the added R stream makes the total parameter count similar.

For discriminators, we use the 70×70 PatchGANs [11] which were also used in CycleGAN. In all models, we apply spectral normalization to the weights of the discriminators [22], which we found to stabilize the adversarial training.

Training We set the reconstruction coefficients on V and C to be ten times the GAN loss, so $\lambda_v = \lambda_c = 10$. We use a smaller coefficient on the the R reconstruction, since it is a much larger tensor: $\lambda_r = 0.1$. We update the discriminator using generated images drawn randomly from a history buffer of size 50. We use the Adam solver [13], with $\beta_1 = 0.5, \beta_2 = 0.999$, a batch size of 4, and a learning rate of 0.0002. After the reconstruction errors stop descending, we linearly decay the learning rate to zero. In total, training can take up to 300,000 steps, which is approximately 3 days on a single Nvidia GTX 1080 TI. This is slower convergence than a traditional CycleGAN (which takes ~100,000 iterations on our data), likely because the objective is harder to optimize when "cheating" is disallowed.

Simplified settings for synthetic data For the experiments with synthetic data, we use a model with fewer parameters. We implement each generator as a fully-connected network with one hidden layer of 32 units and ReLU activation. We implement each adversarial discriminator as a fully-connected network with one hidden layer of 32 units, and leaky ReLU activation. Our experiments suggest that the discriminators have more than sufficient capacity to correctly learn the distributions of C and V and keep equilibrium with the generators. We use the same training setup as in the real-image experiments, except we set the batch size to 128, and training to convergence takes approximately 60,000 iterations, which is 1 hour on a single GPU.

4. Experiments

In this section, we demonstrate that our method outperforms prior work on (1) **accuracy** of disentanglement, (2) **fidelity** of translation, (3) **coverage** of modes (in multimodal translations). Ground-truth disentanglements do not exist in real image data, so we use a simple synthetic scenario to quantitatively evaluate accuracy, then present real-world qualitative results for fidelity and coverage.

4.1. Disentanglement accuracy

One of our claims is that uncooperative optimization is critical for accurate disentanglement. This is based on the idea that a uncooperative models are less able to find "cheats" which bypass the need for accuracy.

In other words, we need to show that "uncooperative" optimization leads to correctly disentangling R, C from within V, in a setting where "cooperative" optimization fails. We present one in which the ground-truth factors are 1D, and entanglement/disentanglement is simply concatenation/splitting. We find that cooperative optimization is incapable of learning this simple operation, whereas uncooperative optimization succeeds.

Models In this experiment, we use two *identical* models (see the "Simplified settings for synthetic data" in Sec. 3.5), and change only the optimization method: one uses the proposed "uncooperative" optimization, and the other uses the baseline "cooperative" optimization.

Data Since ground-truth latent factors are generally unknown in real data, it is necessary to design synthetic data for this experiment. We define the latent factors C and R to be Gaussian distributions. We generate synthetic entanglements $v_i \in V$ by concatenating a sample $c_i \sim C$ with a sample $r_i \sim R$. Specifically, we draw the elements of C from a 1D Gaussian with $\mu = 2.0, \sigma = 1.0$, and draw the elements of R from a 1D Gaussian with $\mu = -2.0, \sigma = 1.0$. We find that results are not sensitive to dimensionality (except in convergence time), and so present only the simplest version here, setting the dimensionality of both C and R to 1, making the dimensionality of V equal to 2. Note that the R domain is never encountered at training time, except in its entangled form inside V. The task is to recover R, using only disentanglement/entanglement cycles, and unpaired samples of V and C.

Metrics We measure the similarity of the actual R domain (used to generate V samples) to the learned R' domain (disentangled from V samples) using the absolute value of the Pearson correlation coefficient $\rho = |\text{cov}(R, R')/(\sigma_R \sigma_{R'})|$, which equals 1 if the two variables have a totally linear relationship, and is closer to 0 otherwise. This (unlike a
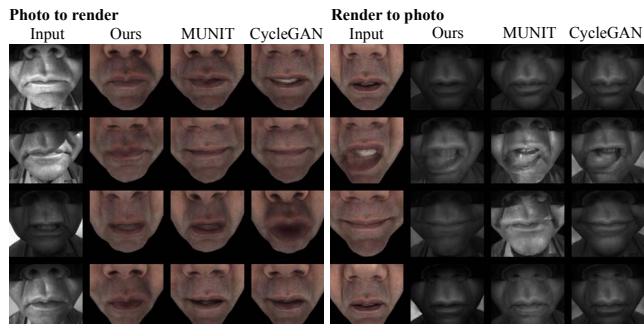

Figure 5: Domain translation results on the face dataset, compared with MUNIT and CycleGAN. In both translation directions, MUNIT and CycleGAN sometimes alter the expression of the subject; our method keeps expression intact.

distance metric) allows solutions where the learned R' is a scaled version of the true R, which is appropriate since scaling may be absorbed in the model weights.

Results Our results are summarized in Figure 4. The two models converge in approximately the same number of iterations. At the end of training, the cooperative version achieves a correlation coefficient of 0.695, while the uncooperative version achieves 0.998. Results vary slightly across iterations (and across initializations), but correlation does not noticeably improve for the cooperative version, even if training is extended to 200k iterations.

Overall, this shows that uncooperative optimization leads the model to disentangle the *true* latent factors, while cooperative optimization does not.

4.2. High-fidelity translation

One of our claims is that the uncooperative training leads to *high-fidelity* translations. By this, we mean that the translation retains as much information as possible from the input, without altering it. To evaluate this, we compare our compare our model's *forward translations* against those of CycleGAN and MUNIT.

Baselines CycleGAN is a popular baseline in unsupervised (but unimodal) image-to-image translation; our architecture is based on it. MUNIT is a state-of-the-art unsupervised *multimodal* image-to-image translation method.

Data We note that MUNIT was originally applied to translating between widely different domains, *e.g.*, translating dogs to lions. While this type of translation is impressive, it is also difficult to evaluate, and it is not clear that close pixel-wise correspondence/fidelity is even desirable in that task.

In this paper, we primarily focus on translating a human face across two appearance domains: photos of the

face captured by a head-mounted camera, and renders of the face produced by a parametric face model (already adapted to the input face). This has an application in social virtual/augmented reality (VR/AR), where we would like users to interact with each other "face-to-face" (inside the virtual environment) as naturally as possible.

We collected the face data ourselves. The real photos (representing the V domain) were captured by a camera attached to the actor's headset, with the lens pointed toward the bottom half of the actor's face; lighting variation was achieved with a set of lights surrounding the actor; background variation was achieved by placing large computer monitors behind the actor and displaying random images. Rendered images of the same face (representing the C domain) were produced by fitting a deep parametric face model to the actor [19], and generating random expressions from a viewpoint similar to the headset view. There are 7074 real photos, and 1000 rendered images. The task is to translate a photo of a face to (or from) a rendered-like image of the same face, while maintaining the face's expression.

For completeness, we also show results on translating architectural facades $\leftrightarrow$ labels [27], which is a task used in prior work [29]. We have also experimented with the aerial photos $\leftrightarrow$ Google maps task [29], but did not find noticeable differences between the methods on that task.

Metrics In the face image experiments—which are necessarily qualitative—we rely on the fact that humans are extremely adept at reading faces [5], and attempt to demonstrate that our model achieves obviously better disentanglements than prior methods. The results on aerial and facade data (introduced in prior work) is harder to interpret at a glance, but close inspection can reveal differences in sharpness and spatial consistency with the input. We note that even when ground truth translations exist, it does not make sense to evaluate against them, since these are many-to-one/one-to-many mappings, and totally unsupervised models (as considered here) cannot be expected to generate labels that match the ground truth (*e.g.*, as assumed in the "FCN score" used in Pix2Pix [11] and CycleGAN [29]).

Results Figure 5 compares our method against MUNIT and CycleGAN on the face dataset. The results show that while CycleGAN and MUNIT perform the appearance translation, they make small but very noticeable shifts in the facial expression, *e.g.*, turning a closed mouth into a smile, or changing a grimace to a pout. This is due to the drawbacks of cooperative training, described earlier. Our method does not have this problem, and translates the faces across domains without altering expression. Figure 6 shows the same experiment but for the facades $\leftrightarrow$ labels task, with similar results: while our method retains, for instance, exact
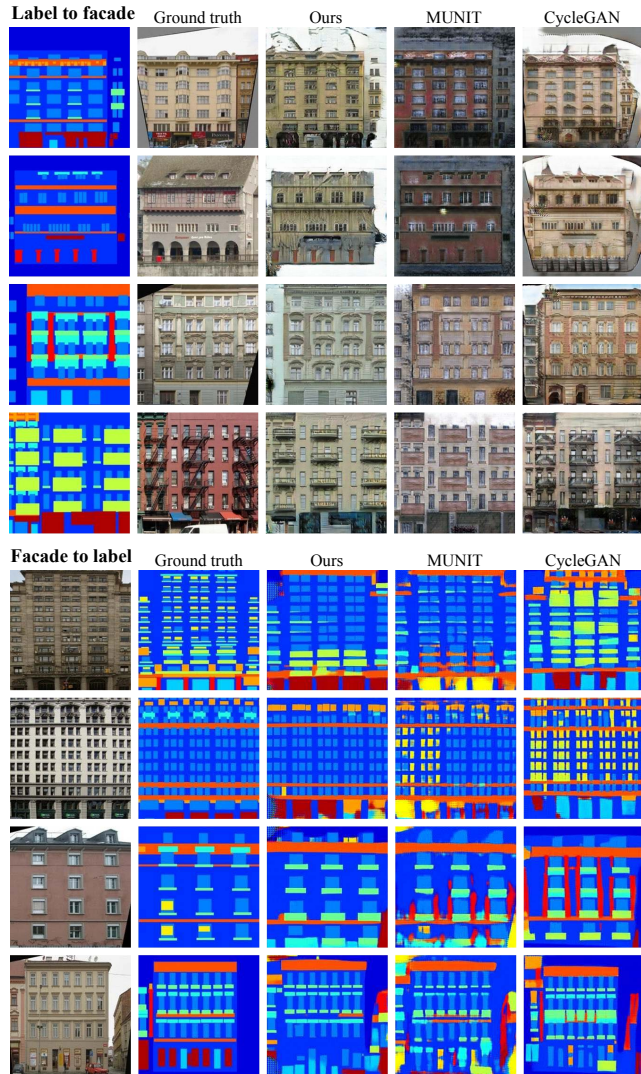

Figure 6: Domain translation results on facade/label images. While MUNIT and CycleGAN introduce artifacts (which make back-translation easier during training), our model performs high-fidelity translation.

spatial positions of the features in either domain, the baseline methods tend to make small shifts in position and scale.

4.3. Multi-modal outputs

Our model is designed to produce multi-modal outputs, through a "mix-and-match" method, where we use C from one input and R from another input, and entangle these to form a novel sample of V. We compare against MUNIT, which is the current state-of-the-art method for this task.

More specifically, generating multiple outputs from a single input involves the following steps: (1) given v_i as input, generate c_i; (2) given an unrelated v_j as input, generate r_j; (3) entangle c_i, r_j, to produce the composite v_{ij}. In the face context, since the domain C contains expression but

not lighting, this setup means extracting expression from one image, and extracting everything else (which is mostly lighting and backgrounds) from another image, and combining these factors into a new image. The experimental setup is similar for MUNIT: a "content code" is generated from v_i, and a "style code" is generated from v_j, and these are encoded into the final output. We do this for multiple v_j, to show the effect of transferring a variety of residual factors onto the same face.

Data We use the same face data as in the high-fidelity task, and also aerial photos $\leftrightarrow$ Google maps [29], which we find has more evident multi-modality than the facades.

Results Figure 7 shows the results of this experiment on the faces dataset, for MUNIT and our model. The figure shows expressions from v_i across rows, and residuals from v_j (*i.e.*, lighting/background conditions) across columns. For an overview of the results, the reader may scan across rows to inspect that expression is transferred from the leftmost row, and scan across columns to inspect that lighting and backgrounds are transferred from the topmost row. MUNIT appears to have only learned to transfer the global intensity from the v_j source. Our model appears to be transferring backgrounds, and even casting distinct shadows onto the face. However, some shadows appear reduced in intensity (*e.g.*, third column), suggesting that expression-lighting disentanglement is not perfect here.

In the supplementary, we also show results of this experiment on the aerial photos $\leftrightarrow$ Google maps dataset, where we treat the Google map as C (assuming it has less information), and the aerial photos as V. In this domain, it appears MUNIT transfers very little from the residual, while our model incorporates textures and objects (*e.g.*, note the white object transferred from the first residual). Both methods appear to retain the spatial layout of the input map.

5. Discussion

In this work, we address the *compensation* issue in translation cycle-consistency, which typically diminishes the utility of the reconstruction loss. In compensation, the back-translator (undesirably) adapts to the weaknesses and shortcuts of the forward-translator. Hypothetically, there is another way to (partially) defeat the loss, which may be called *exploitation*. In exploitation, the forward-translator (undesirably) adapts to the weaknesses and shortcuts of the back-translator. The enduring exploitation issue may explain the subtle imperfections in our outputs.

Another limitation of our approach is that we do not address *many-to-many* mappings. Our approach is only multimodal in one direction.
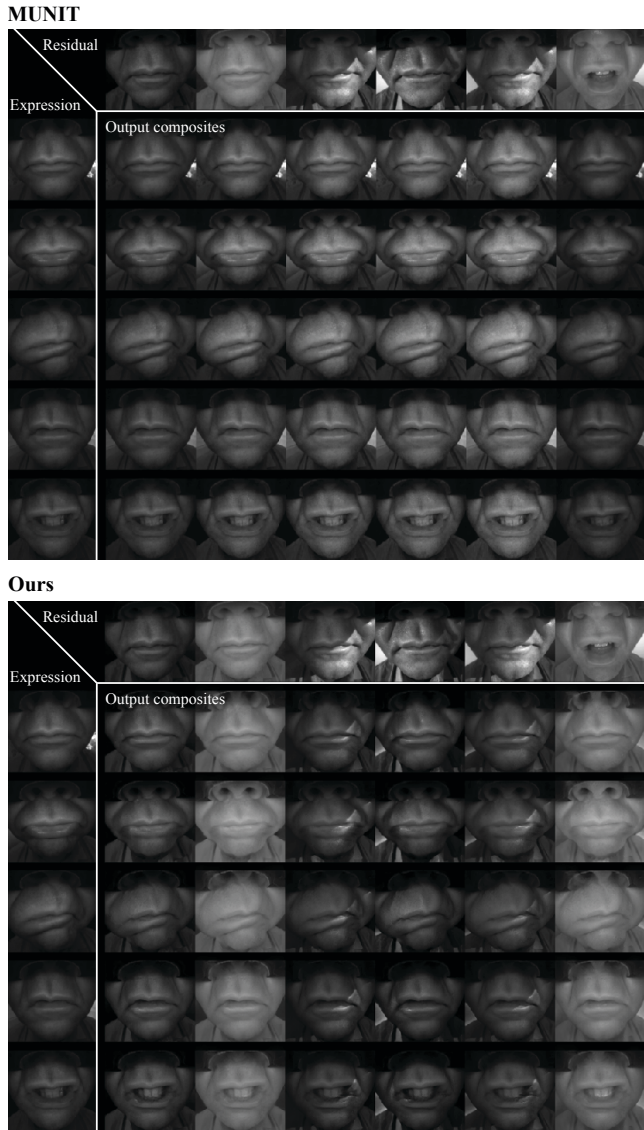
MUNIT

Ours

Figure 7: Face relighting results of MUNIT (top) vs our model (bottom). In each table, the leftmost column shows the input c_i from which expression is drawn; the top row shows the input v_i from which everything else is drawn.

In summary, we introduced the problem of high-fidelity image-to-image translation, motivated it for augmented reality applications, and presented an unsupervised method for solving it. We identified a fundamental cause of low-fidelity translations: cooperation between the forward translator and the backward translator, which allows the forward-translation to "hide" information, and the back-translator to "recover" from noticeable errors. This is a critical problem in real applications. We presented an "uncooperative" optimization scheme that prevents the problem. Our results demonstrate that uncooperative optimization leads to high-fidelity image translations, making image-to-image translation not only fun, but useful for augmented reality.

References

[1] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. C. Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. In *ICML*, 2018. 2, 3

[2] I. Belghazi, S. Rajeswar, A. Baratin, R. D. Hjelm, and A. Courville. MINE: Mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018. 3

[3] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016. 3

[4] C. Chu, A. Zhmoginov, and M. Sandler. CycleGAN: A master of steganography. *arXiv preprint arXiv:1712.02950*, 2017. 2

[5] P. Ekman. *The face of man: Expressions of universal emotions in a New Guinea village*. Garland Publishing, Incorporated, 1980. 4, 7

[6] H. Fu, M. Gong, C. Wang, K. Batmanghelich, K. Zhang, and D. Tao. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. *arXiv preprint arXiv:1706.00826*, 2018. 2

[7] A. Gonzalez-Garcia, J. van de Weijer, and Y. Bengio. Image-to-image translation for cross-domain disentanglement. *arXiv preprint arXiv:1805.09730*, 2018. 2, 3

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[9] N. Hadad, L. Wolf, and M. Shahar. A two-step disentanglement method. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[10] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. *arXiv preprint arXiv:1804.04732*, 2018. 2, 3

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017. 2, 5, 7

[12] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 5

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[14] J. Kossaifi, L. Tran, Y. Panagakis, and M. Pantic. Gagan: Geometry-aware generative adversarial networks. *arXiv preprint arXiv:1712.00684*, 2017. 2

[15] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, pages 2539–2547, 2015. 3

[16] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang. Diverse image-to-image translation via disentangled representations. *arXiv preprint arXiv:1808.00948*, 2018. 2, 3

[17] K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *CV*, pages 991–999, 2015. 3

[18] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017. 2, 3

[19] S. Lombardi, J. Saragih, T. Simon, and Y. Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics (TOG)*, 37(4):68, 2018. 7

[20] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2813–2821. IEEE, 2017. 3

[21] M. Mirza and S. Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. 2

[22] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 5

[23] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. GANerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59, 2018. 2

[24] S. Reed, K. Sohn, Y. Zhang, and H. Lee. Learning to disentangle factors of variation with manifold interaction. In *International Conference on Machine Learning*, pages 1431–1439, 2014. 3

[25] Z. Shu, M. Sahasrabudhe, A. Guler, D. Samaras, N. Paragios, and I. Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. *arXiv preprint arXiv:1806.06503*, 2018. 3

[26] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5444–5453. IEEE, 2017. 3

[27] R. Tyleček and R. Šára. Spatial pattern templates for recognition of objects with regular structure. In *German Conference on Pattern Recognition*, pages 364–374. Springer, 2013. 7

[28] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with pixelcnn decoders. *CoRR*, abs/1606.05328, 2016. 2

[29] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *CVPR*, 2017. 1, 2, 5, 7, 8

[30] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *Advances in neural information processing systems*, pages 465–476, 2017. 2, 3