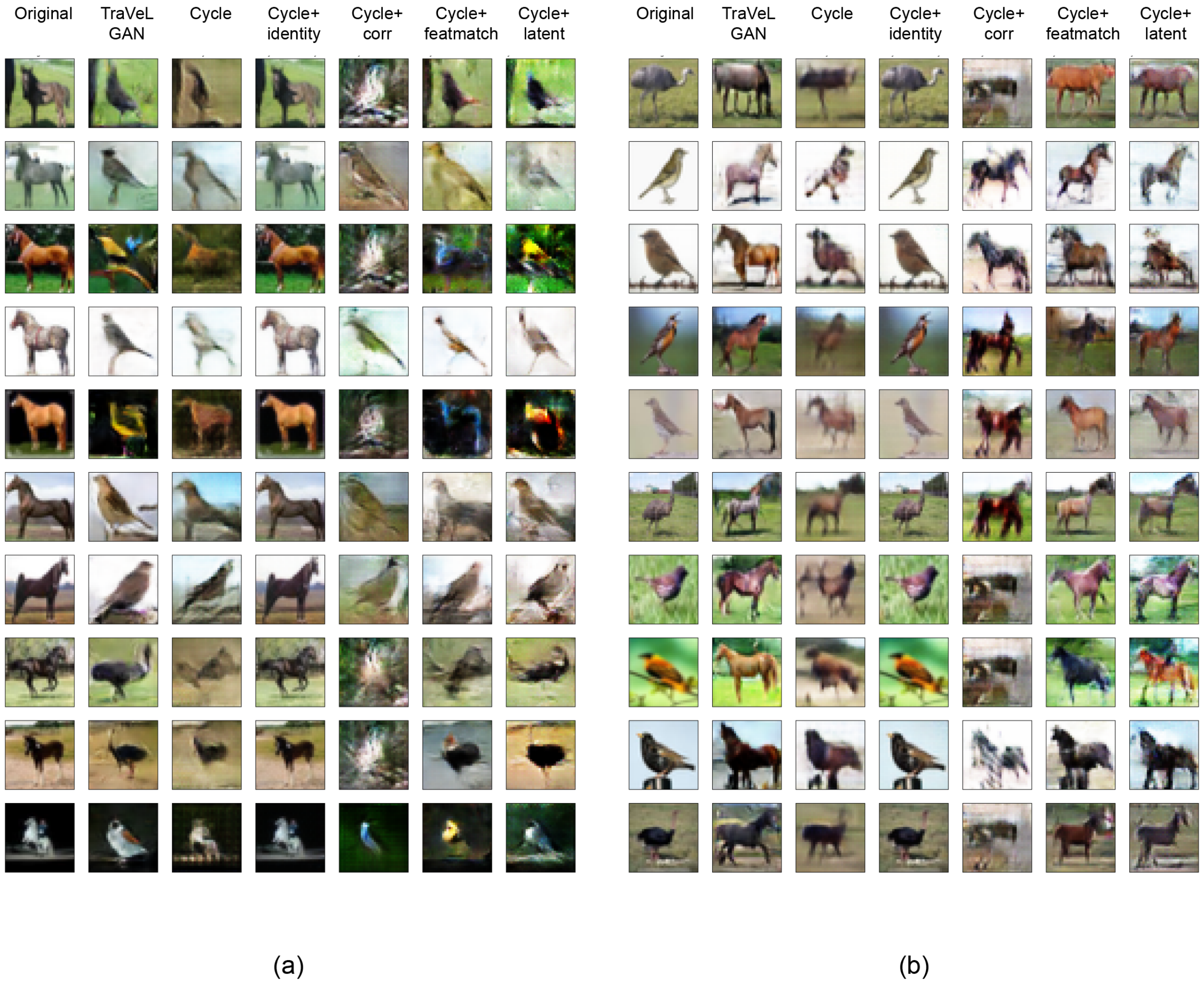

Figure S1: Learning to map between two CIFAR domains: (a) horse to bird (b) bird to horse.

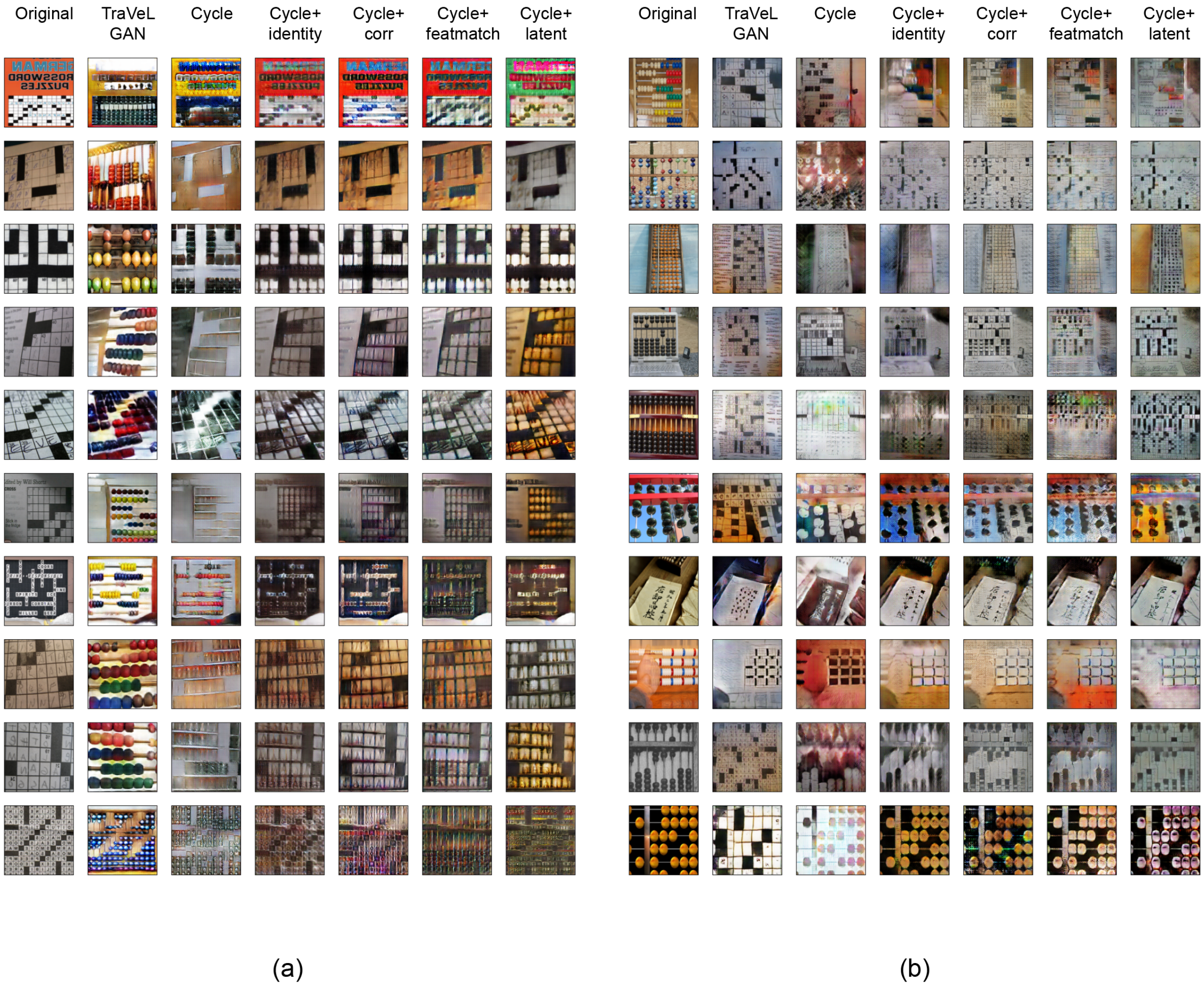

Figure S2: Learning to map between two imagenet domains: (a) crossword to abacus (b) abacus to crossword.

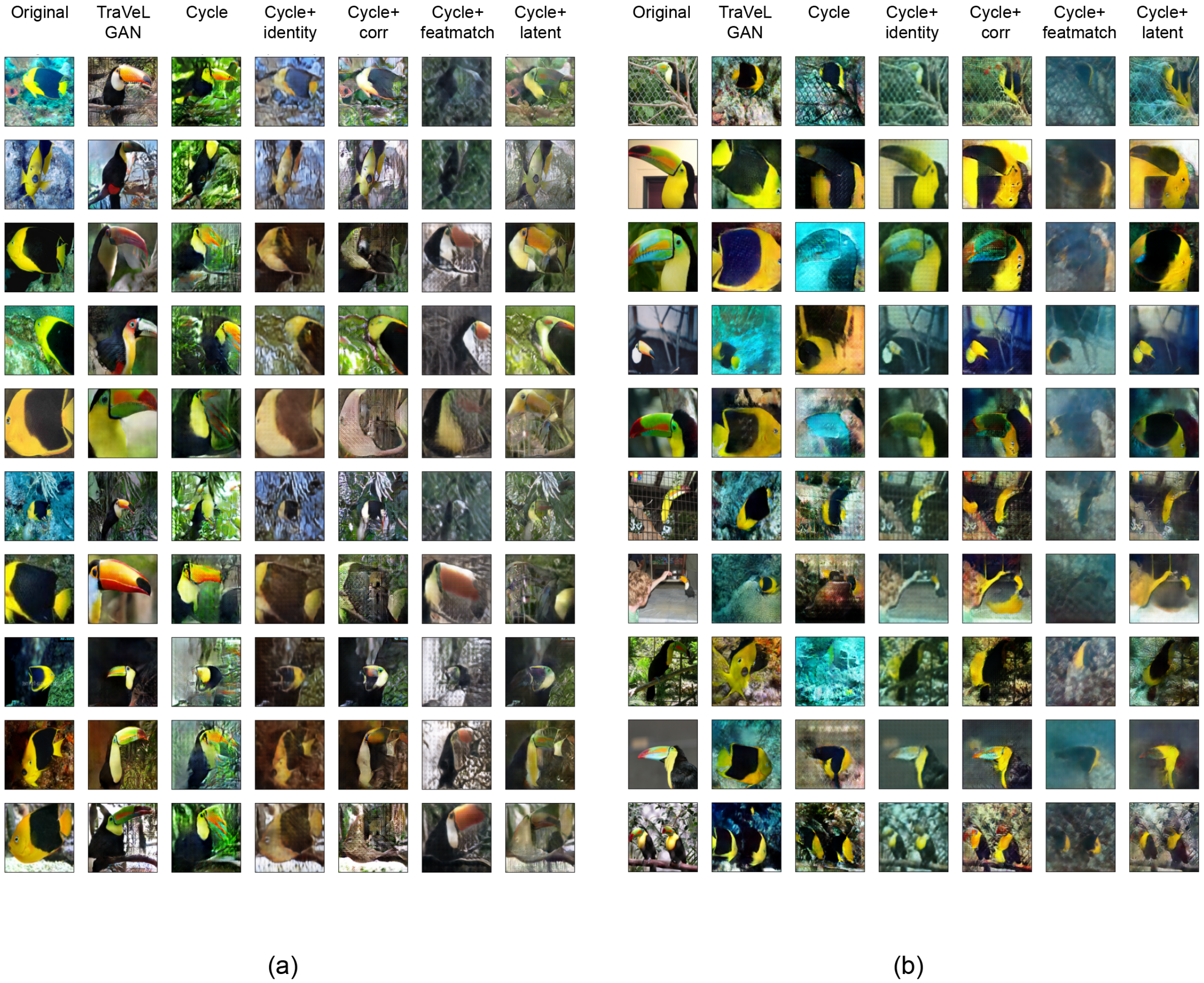

Figure S3: Learning to map between two imagenet domains: (a) rock beauty to toucan (b) toucan to rock beauty.

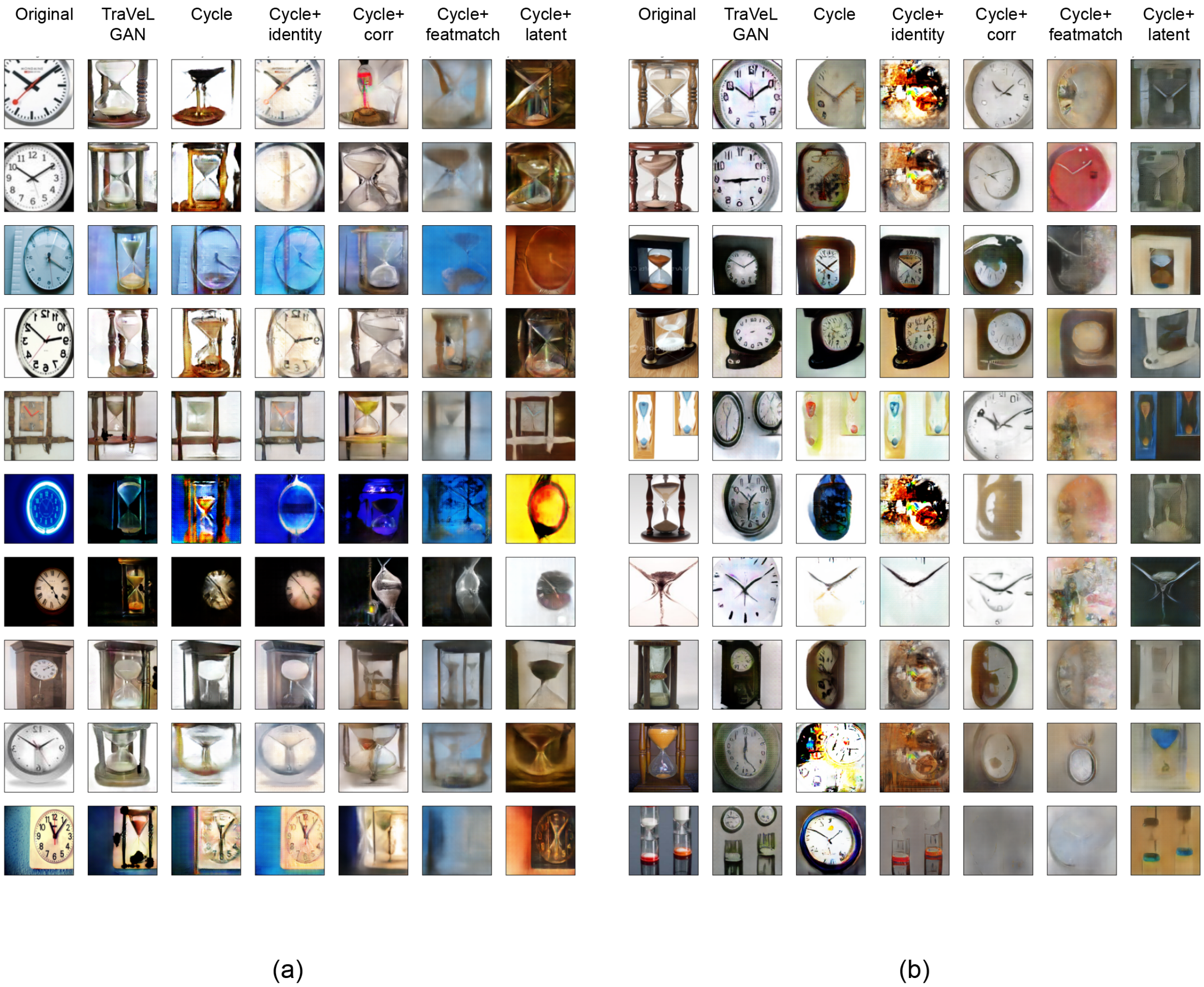

Figure S4: Learning to map between two imagenet domains: (a) clock to hourglass (b) hourglass to clock.

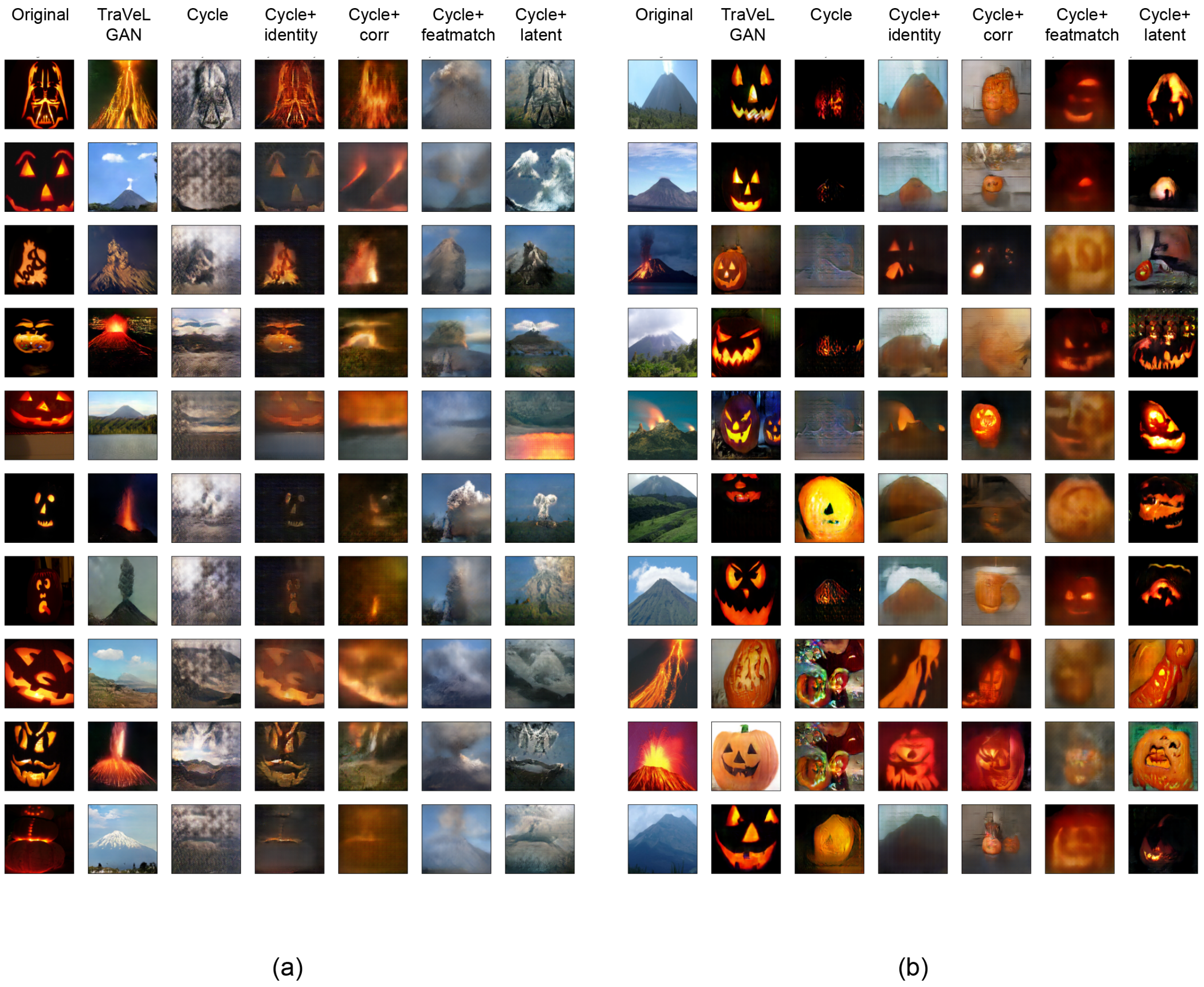

Figure S5: Learning to map between two imagenet domains: (a) jack-o-lantern to volcano (b) volcano to jack-o-lantern.

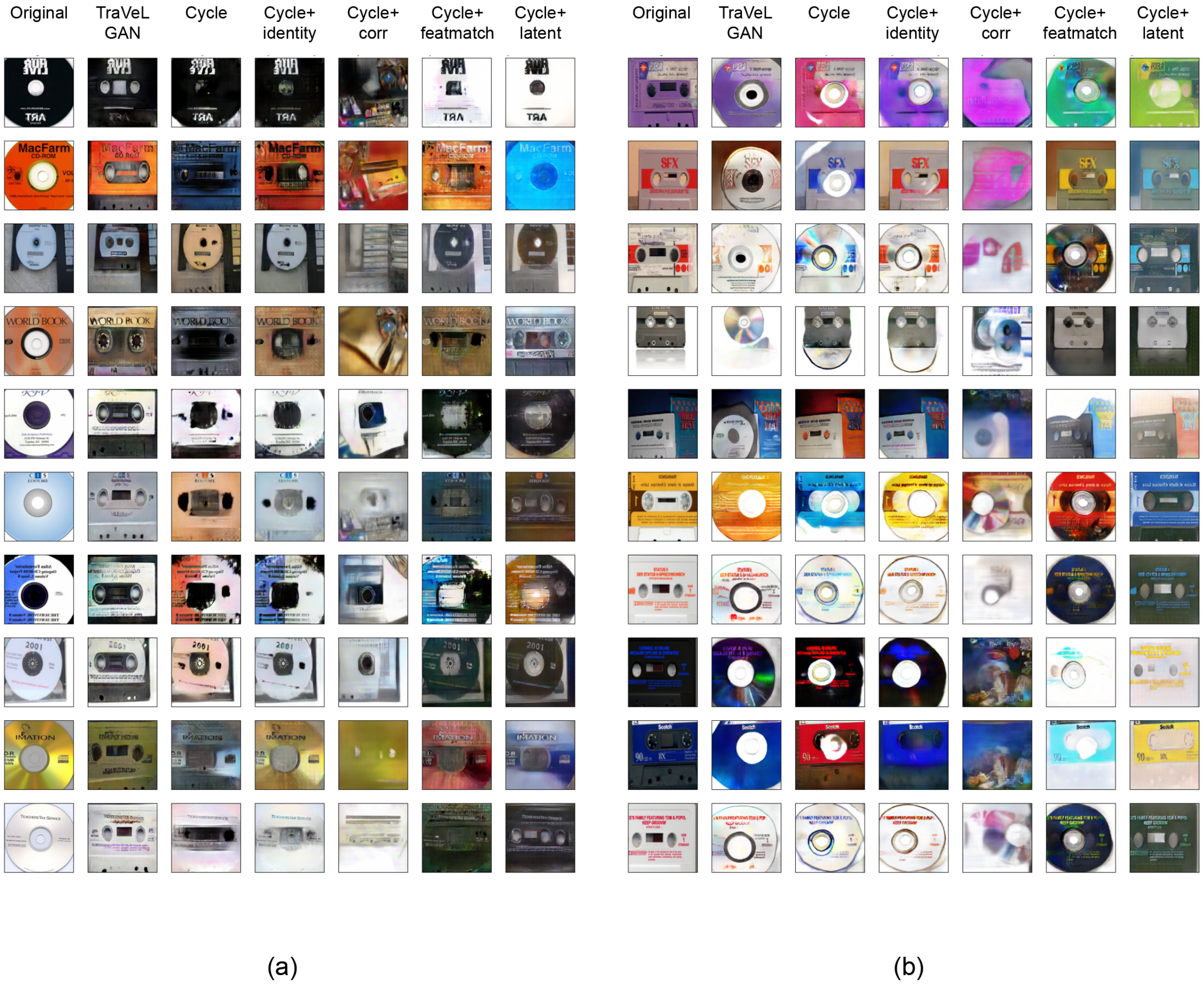

Figure S6: Learning to map between two imagenet domains: (a) cd to cassette (b) cassette to cd.

Quantitative results Quantitative results are summarized by the FID score (Table 3) and the discriminator score (Table 4). We note that these scores were both designed to evaluate models that attempt to generate the full diversity of the Imagenet dataset, while in our case we only map to a single class.

The Fréchet Inception Distance (FID score) [12] calculates the Fréchet distance between Gaussian models of the output of a the pre-trained Inception network [34] on real and generated images, respectively. Lower distances indicate better performance. The results are the mean of the scores from each direction.

The discriminator score is calculated by training a new discriminator, distinct from the one used during training, to distinguish between real and generated images in a domain. A score of zero means the discriminator was certain every generated image was fake, while higher scores indicate the generated images looked more like real images. As in the FID, the results are the mean of the scores from each direction.

Optimization and training parameters Optimization was performed with the adam [21] optimizer with a learning rate of 0.0002, $\beta_1 = 0.5$, $\beta_2 = 0.9$. Gradient descent was alternated between generator and discriminator, with the discriminator receiving real and generated images in distinct batches.

Architecture The TraVeLGAN architecture is as follows. Let d denote the size of the image. Let c_n be a standard stride-two convolutional layer with n filters, t_n be a stride-two convolutional transpose layer with kernel size four and n filters, and f_n be a fully connected layer outputting n neurons. The discriminator D has layers until the size of the input is four-by-four, increasing the number of filters by a factor of two each time, up to a maximum of eight times the original number (three layers for CIFAR and five layers for Imagenet). This last layer is then flattened and passed through a fully connected layer. The overall architecture is thus $c_n - c_{2n} - c_{4n} - c_{8n} - c_{8n} - f_1$. The siamese network has the same structure as the discriminator except its latent space has size 1000, yielding the architecture $c_n - c_{2n} - c_{4n} - c_{8n} - c_{8n} - f_{1000}$. The generator uses the U-Net architecture [30] that has skip connections that concatenate the input in the symmetric encoder with the decoder, yielding layers of $c_n - c_{2n} - c_{4n} - c_{4n} - c_{4n} - t_{8n} - t_{8n} - t_{8n} - t_{4n} - t_{2n} - t_3$. For the cycle-consistency networks, the architectures of the original implementations were used, with code from [39], [39], [3], [20], for the cycle, cycle+identity, cycle+corr, and cycle+featmatch, respectively. All activations are leaky rectified linear units with leak of 0.2, except for the output layers, which use sigmoid for the discriminator, hyperbolic tangent for the generator, and linear for the siamese network. Batch normalization is used for every layer except the first layer of the discriminator. All code was implemented in Tensorflow [1] on a single NVIDIA Titan X GPU.

CIFAR While the CIFAR images [22] are relatively simple and low-dimensional, it is a deceptively complex task compared to standard domain mapping datasets like CelebA, where they are all centered close-ups of human faces (i.e. their shoulders or hair are in the same pixel locations). The cycle-consistent GANs struggle to identify the characteristic shapes of each domain, instead either only make small changes to the images or focusing on the color tone. The TraVeLGAN, on the other hand, fully transfers images to the target domain. Furthermore, the TraVeLGAN preserves semantics like orientation, background color, body color, and composition in the pair of image (complete comparison results in Figure S1)

Interpretability As the siamese latent space is learned to preserve vector transformations between images, we can look at how that space is organized to tell us what transformation the network learned at a dataset-wide resolution. Figure S7 shows a PCA visualization of the siamese space of the CIFAR dataset with all of the original domain one (bird) and domain two (horse) images. There we can see that S learned a logical space with obvious structure, where mostly grassy images are in the bottom left, mostly sky images in the top right, and so forth. Furthermore, the layout is analogous between the two domains, verifying that the network automatically learned a notion of similarity between the two domains. We also show every generated image across the whole dataset in this space, where we see that the transformation vectors are not just interpretable for some individual images and not others, but are interpretable across the entire distribution of generated images.

Salience We next perform a salience analysis of the TraVeL loss by calculating the magnitude of the gradient at each pixel in the generated image with respect to each pixel in the original image (Figure S8). Since the TraVeL loss, which enforces the similarity aspect of the domain mapping problem, is parameterized by another neural network S, the original image contributes to the generated image in a complex, high-level way, and as such the gradients are spread richly over the entire foreground of the image. This allows the generator to make realistic abacus beads, which need to be round and shaded, out of square and uniform pixels in the crossword. By contrast, the cycle-consistency loss requires numerical precision in the pixels, and as such the salience map largely looks like a grayscale version of the real image, with rigid lines and large blocks of homogeneous pixels still visible. This is further evidence that the cycle-consistency

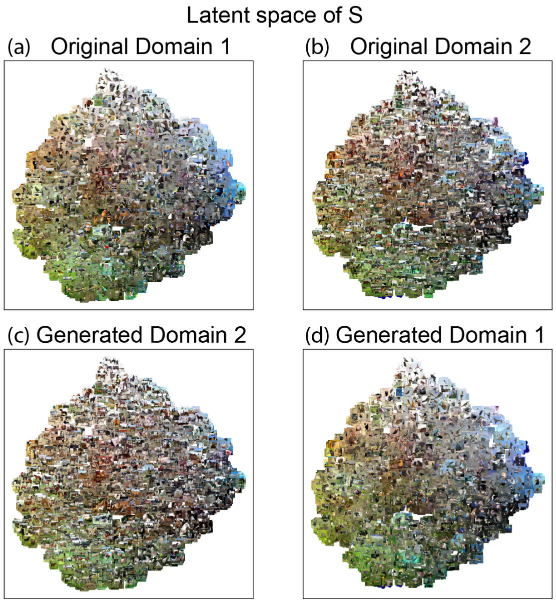

Figure S7: Having access to the siamese space output by S provides an interpretability of the TraVeLGAN's domain mapping that other networks lack. PCA visualizations on the CIFAR example indicate S has indeed learned a meaningfully organized space for G to preserve transformation vectors within.

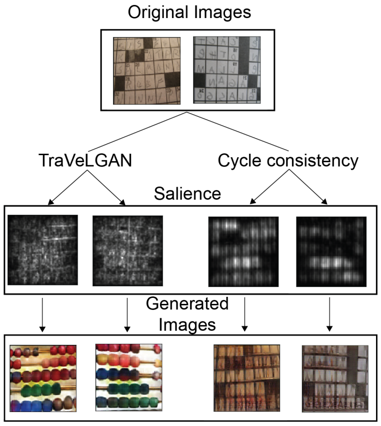

Figure S8: A salience analysis exploring how the TraVeLGAN's objective loosens the restriction of cycle-consistency and allows it more flexibility in changing the image during domain transfer. The TraVeL loss requires significantly less memorization of the input pixels, and as a result, more complex transformations can be learned.

loss is preventing the generator from making round beads with colors that vary over the numerical RGB values.