Representational Disentanglement for Multi-Domain Image Completion

Anonymous authors
Paper under double-blind review

Abstract

Multi-domain data are widely leveraged in vision applications to take advantage of complementary information from each modality, e.g., brain tumor segmentation from multi-parametric magnetic resonance imaging (MRI). However, due to different imaging protocol and data loss or corruption, the availability of images in all domains could vary amongst multiple data sources in practice, which makes it challenging to train and test a universal model with a varied set of input data. To tackle this problem, we propose a general approach to complete the possible missing domain of the input data in a variety of application settings. Specifically, we develop a novel generative adversarial network (GAN) architecture that utilizes a representational disentanglement scheme for shared 'skeleton' encoding and separate 'flesh' encoding across multiple domains. We further illustrate that the learned representation in the multi-domain image translation could be leveraged for higher-level recognition, like segmentation. Specifically, we introduce a unified framework of image completion branch and segmentation branch with a shared content encoder. We demonstrate constant and significant performance improvement by integrating the proposed representation disentanglement scheme in both multi-domain image completion and image segmentation tasks using three evaluation datasets individually for brain tumor segmentation, prostate segmentation, and facial expression image completion.

1 Introduction

Multi-domain images are often required as inputs due to the nature that different domains could provide complementary knowledge in various vision tasks. For example, four medical imaging modalities, MRI with T1, T1-weighted, T2-weighted, FLAIR (FLuid-Attenuated Inversion Recovery), are acquired as a standard protocol to segment the accurate tumor regions for each patient in the brain tumor segmentation task (Menze et al., 2014). Each modality can provide distinct features to locate the true tumor boundaries from a differential diagnosis perspective. Additionally, when it comes to the natural image tasks, there are similar scenarios such as person re-identification across different cameras and different times (Zheng et al., 2015; 2019). Here, the medical images in different contrast modalities or natural images with the person under different appearance or cameras can both be considered as different domains, which all contribute to depict the underlying object or scene from different aspects of view.

However, some domains might be missing in practice. In large-scale datasets from multiple institutes, it is generally difficult or even infeasible to guarantee the availability of complete domains for all the data. For example, in some cases, the patients might lack some imaging scans due to different imaging protocol or data loss or corruption. In terms of taking the most advantage of all these rare and valuable data, it is costly to just throw away the incomplete samples during training, and even infeasible to test with missing domain input. Thus, it becomes necessary to effectively artificially generate the missing data. An intuitive approach is to substitute the missing domains with the nearest neighbor among other existing samples, but this might lack of semantic consistency among domains of the input sample since it only focuses on pixel-level similarity with existing data.

Additionally, the recent success of GANs (Goodfellow et al., 2014; Mirza & Osindero, 2014; Isola et al., 2017; Zhu et al., 2017a;b; Kim et al., 2017; Liu et al., 2017; Choi et al., 2018; Yoon et al., 2018; Lee et al., 2019a) in image-to-image translation provides another possible solution for this

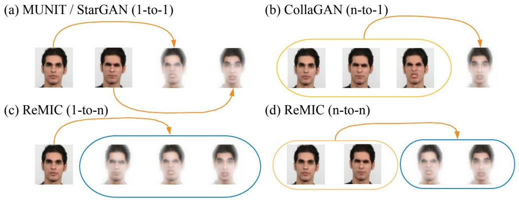

Figure 1: Image translation tasks using (a) StarGAN and MUNIT (1-to-1), (b) CollaGAN (n-to-1), (c) ReMIC (1-to-n), and (d) ReMIC (n-to-n). ReMIC completes the missing domain images that can be randomly distributed (N-to-n, $N \in \{1, ..., n\}$) in the input set. It makes our approach a more general and flexible framework for image translation tasks.

challenge. By training a GAN-based model to generate images for missing domains, we are more likely to keep the semantic consistency with existing domains by learning the semantic representations. CycleGAN (Zhu et al., 2017a) shows an impressive performance in the image-to-image translation task via cycle-consistency constraints between real images and generated images. However, CycleGAN mainly focuses on the 1-to-1 mapping between two specific domains and assumes the corresponding images in two domains strictly share the same representation in latent space. This is limited in multi-domain applications since $n(n-1)$ mapping functions are required if there are n domains. Following this, StarGAN (Choi et al., 2018) proposes a general method for multi-domain image translation using a mask vector in inputs to specify the desired target domain. Meantime, RadialGAN (Yoon et al., 2018) also deals with the multi-domain generation problem with the assumption that all the domains share the same latent space. Although StarGAN and RadialGAN make it possible to generate images in different target domains through the 1-to-n translation, the representation learning and image generation are always conditioned on the input with only one source domain. In order to take advantage of multiple available domains as inputs for representation learning, CollaGAN (Lee et al., 2019a) proposes a collaborative model to incorporate multiple available domains to generate one missing domain. Similar to StarGAN, CollaGAN only relies on the cycle-consistency constraints to preserve the contents in generated images, which are actually implicit constraints between real images and fake images in pixel level. Additionally, since the target domain is controlled by an one-hot mask vector in the input, CollaGAN is essentially doing n-to-1 translation in one inference. Our goal in this work is to propose a more general n-to-n image translation framework that could overcome the aforementioned limitations as illustrated in the Fig. 1.

Recently, learning disentangled representation is proposed to capture the full distribution of possible outputs by introducing a random style code (Chen et al., 2016; Higgins et al., 2017; Huang et al., 2018; Lee et al., 2018). InfoGAN (Chen et al., 2016) and β-VAE (Higgins et al., 2017) learn the disentangled representation without supervision. In image-to-image translation tasks, DIRT (Lee et al., 2018) learns disentangled content and attribute features by exchanging the features encoded from images of two domains respectively and then reverting back again. The image consistency during translation is constrained by code reconstruction and image reconstruction. With a similar code exchange framework, MUNIT (Huang et al., 2018) assumes a prior distribution on style code, which allows sampling different style codes to generate multiple images in target domain. However, both DIRT and MUNIT only deal with image translation between two domains, which is not efficient to train $n(n-1)$ mapping functions in a n-domain task.

Inspired by previous works, we propose a n-to-n multi-domain image translation approach based on the representational disentanglement scheme for multi-domain image completion (ReMIC). Specifically, our contributions are three-fold: (1) We propose a GAN architecture with representation disentanglement to learn domain-shared features and domain-specific features for the more general and flexible n-to-n image translation; (2) We illustrate the effectiveness of the representation learning for high-level tasks by building a unified framework for jointly learning the generation and segmentation with a shared content encoder; (3) Extensive experiments with three different datasets demonstrate that the proposed method achieves better performance than previous state-of-the-art methods on both generation and segmentation.

2 Method

As discussed above, images from different domains for the same data sample could present their exclusive features of the data subject. Nonetheless, they also inherit some global content structures

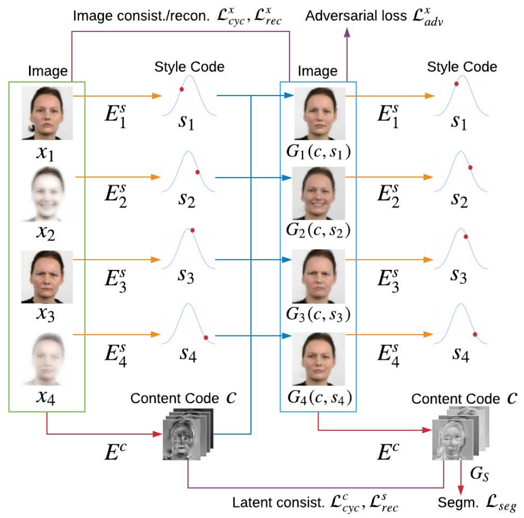

Figure 2: Overview of the proposed n-to-n multi-domain completion and segmentation framework. $n = 4$ and two domain data are missing in this example. Our model contains a unified content encoder E^c (red lines), domain-specific style encoders E^s_i ($1 \leq i \leq n$, orange lines) and generators G_i ($1 \leq i \leq n$, blue lines). A variety of loss was adopted (burgundy lines), i.e., image consistency loss for visible domains, latent consistency loss, adversarial loss and reconstruction loss for the generated images. Furthermore, the representational learning framework is flexible to combine a segmentation generator G_S following the content code for the unified image generation and segmentation.

of it. Let us take the parametric MRI for brain tumors as an example. T2 and FLAIR MRI can highlight the differences in tissues' water relaxational properties, which will distinguish the tumor tissue from normal ones. Contrasted T1 MRI can examine the pathological intratumoral take-up of contrast agents so that the boundary between tumor core and the rest will be highlighted. However, the underline anatomical structure of the brain is shared by all these three modalities. With the availability of multiple domain data, it will be meaningful to decompose the images into the shared content structure (skeleton) and meanwhile distinguish and model their unique characteristics (flesh) through learning. Therefore, we will be able to reconstruct the missing image during the testing by using the shared skeleton (extracted from the available data domains) and a sampled flesh from the learned model. Without assuming a fixed number of missing domains during the model training, the learned framework could flexibly handle one or more missing domains. In addition, we further enforce the accuracy of the extracted content structure by connecting it to the segmentation tasks. In such manner, the disentangled representations of multiple domain images (both the skeleton and flesh) can help in both the image completion and segmentation tasks.

Suppose there are n domains: $\{\chi_\mathbf{1}, \chi_\mathbf{2}, \cdots, \chi_\mathbf{n}\}$. Let $x_1 \in \chi_\mathbf{1}$, $x_2 \in \chi_\mathbf{2}$, $\cdots$, $x_n \in \chi_\mathbf{n}$ are images from n different domains respectively, which are grouped data describing the same subject $\mathbf{x} = \{x_1, \cdots, x_n\}$. In total, we assume the whole dataset contains M independent data samples. For each data sample, we assume one or more of the n images might be randomly missing. The goal of our first task is to complete all the missing domains for all the samples.

To accomplish the completion of all missing domains from a random set of available domains, we assume the n domains share the latent representation of underline structure. We name the shared latent representation as *content code* and meanwhile each domain also contains the domain-specific latent representation, i.e., *style code* that is related to the various features or attributes in different domains. The missing domains can be reconstructed from these two-aspect of information through the learning of deep neural networks. Similar to the setting in MUNIT (Huang et al., 2018), we also assume a prior distribution for style latent code as $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ to capture the full distribution of possible outputs in each domain. However, MUNIT trains separate content encoder for each domain and enforce the disentanglement via coupled cross-domain translation during training while the proposed method employs one single content encoder to extract the true representation shared across all the domains.

2.1 Unified Image Completion and Segmentation Framework

As shown in Figure 2, our model contains a unified content encoder E^c and domain-specific style encoders E^s_i ($1 \leq i \leq n$). Content encoder E^c extracts the content code c from all existing domains: $E^c(x_1, x_2, \cdots, x_n) = c$. For the missing domains, we use zero padding in corresponding input channels. For each domain, a style encoder E^s_i learns the domain-specific style code s_i from the corresponding domain image x_i respectively ($1 \leq i \leq n$): $E^s_i(x_i) = s_i$.

During the training, our model captures the content code c and style codes s_i $(1 \leq i \leq n)$ (assumed in a prior distribution of $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ as shown in Figure 2) through the encoding process (denoted as red and orange arrows respectively) with a random set of input images (in green box). We only need to train a single ReMIC model to generate all these missing images from sets of available domains.

In the generation process, our model samples style codes from prior distribution and recombines content code to generate images in n domains through generators G_i $(1 \leq i \leq n)$ (denoted as blue arrows). The generator G_i for each domain generates the fake images in corresponding domain from the domain-shared content code and the domain-specific style code: $G_i(c, s_i) = \tilde{x}_i$.

As discussed before, it is fairly common to find that there are missing domains for input data. Some straight-forward solutions to complete all the missing images include zero filling, average image, or generating images via image translation model, etc. Alternatively, based on the content code learned in our model, we could develop a unified model for multi-task learning of both generation and segmentation. Specifically, another branch of segmentation generator G_S is added after content encoder to generate segmentation mask. By optimizing the generation loss and segmentation Dice loss (detailed in Section 2.2) simultaneously, the model could adaptively learn how to generate missing images to promote the segmentation performance.

2.2 Training Loss

In the training of GAN models, the setting of losses is of paramount importance to the final generation results. Our loss objective contains the cycle-consistency loss of images and codes within and across domains, adversarial loss and reconstruction loss on the generation and input images.

Image Consistency Loss: For each sample, the content and style encoders are able to extract a domain-shared content code and domain-specific style codes respectively for each available domain. Then by recombining the content and style codes, the domain generators are expected to reconstruct the input images from each domain. The image consistency loss is defined to constrain the reconstructed images and real images as in the flow chat of "$Image \rightarrow Code \rightarrow Image$".

$$\mathcal{L}^{x_i}_{cyc} = \mathbb{E}_{x_i \sim p(x_i)}[\| \; G_i(E^c(x_1, x_2, \cdots, x_n), E^s_i(x_i)) - x_i \; \|_1] \tag{1}$$

where $p(x_i)$ is the data distribution in domain χ_i $(1 \leq i \leq n)$.

Latent Consistency Loss: Next, the generated fake images can also be encoded as content and style codes by using the same encoders. The latent consistency loss constrains the code before decoding and after encoding again in the direction of "$Code \rightarrow Image \rightarrow Code$".

$$\mathcal{L}^{c}_{cyc} = \mathbb{E}_{c \sim p(c), s_i \sim p(s_i)}[\| \; E^c(G_1(c, s_1), G_2(c, s_2), \cdots, G_n(c, s_n)) - c \; \|_1] \tag{2}$$

$$\mathcal{L}^{s_i}_{cyc} = \mathbb{E}_{c \sim p(c), s_i \sim p(s_i)}[\| \; E^s_i(G_i(c, s_i)) - s_i \; \|_1] \tag{3}$$

where $p(s_i)$ is the prior distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $p(c)$ is given by $c = E^c(x_1, x_2, \cdots, x_n)$ and $x_i \sim p(x_i)$ $(1 \leq i \leq n)$.

Adversarial Loss: The adversarial learning between generators and discriminators forces the data distribution of the generated fake images to be close to the real images' distribution for each domain.

$$\mathcal{L}^{x_i}_{adv} = \mathbb{E}_{c \sim p(c), s_i \sim p(s_i)}[\log(1 - D_i(G_i(c, s_i)))] + \mathbb{E}_{x_i \sim p(x_i)}[\log D_i(x_i)] \tag{4}$$

where D_i is the discriminator for domain i to distinguish the generated fake images $\tilde{x}_i$ and real images $x_i \in \chi_i$.

Reconstruction Loss: In addition to the feature-level consistency mentioned above to constrain the relationship between the generated fake images and real images in different domains, we also constrain the pixel-level similarity between generated images and ground truth images in the same domain during training stage.

$$\mathcal{L}^{x_i}_{rec} = \mathbb{E}_{c \sim p(c), s_i \sim p(s_i)}[\| \; G_i(c, s_i) - x_i \; \|_1] \tag{5}$$

Total Loss: The encodes, decoders and discriminators are jointly trained to optimize the total objective as follows.

$$\begin{aligned} &\min_{E^c, E^s_1, \cdots, E^s_n} \max_{D_1, \cdots, D_n} \mathcal{L}(E^c, E^s_1, \cdots, E^s_n, D_1, \cdots, D_n) \\ &= \lambda_{adv} \sum_{i=1}^{n} \mathcal{L}^{x_i}_{adv} + \lambda^x_{cyc} \sum_{i=1}^{n} \mathcal{L}^{x_i}_{cyc} + \lambda^c_{cyc} \mathcal{L}^c_{cyc} + \lambda^s_{cyc} \sum_{i=1}^{n} \mathcal{L}^{s_i}_{cyc} + \lambda_{recon} \sum_{i=1}^{n} \mathcal{L}^{x_i}_{recon} \end{aligned} \tag{6}$$

where λ_{adv}, λ^x_{cyc}, λ^c_{cyc}, λ^s_{cyc}, and λ_{recon} are weights.

In the n-to-n image translation task, the model learns a complementary representation of multiple domains, which can also facilitate the high-level recognition tasks. For example, extracted content code (containing the underline anatomical structures) can largely benefit the segmentation of organs and lesions in medical image analysis. On the other side, the segmentation task can enforce the learning of a more representative content encoder. Therefore, we train a multi-task network for both the segmentation and generation. In the proposed framework, we construct a unified generation and segmentation model by adding a segmentation generator G_S following the content code out from the generated fake images as shown in Fig. 2. We utilize dice loss (Salehi et al., 2017) for accurate segmentation of multiple input images.

$$\mathcal{L}_{seg} = 1 - \frac{1}{L}\sum_{l=1}^{L} \frac{\sum_p \hat{y}_p(l)y_p(l)}{\sum_p \hat{y}_p(l) + y_p(l) - \hat{y}_p(l)y_p(l)} \tag{7}$$

where L is the total number of classes, p is the pixel position in the image, $\hat{y}$ is the predicted segmentation probability map from G_S and y is the ground truth segmentation mask. The segmentation can loss can be added into the total loss in Equation 6 for an end-to-end joint learning.

3 Experimental Results and Discussion

We firstly show the advantage of our proposed method in the n-to-n image completion task given a random set of available domains. Moreover, we illustrate that the proposed model (a variation with two branches) provides an efficient solution to multi-domain segmentation with missing image inputs. To demonstrate the generalizability of the proposed algorithm, we evaluate the proposed method on a natural image dataset as well as two medical image datasets.

Totally three datasets are employed in our experiments, i.e., BraTS, ProstateX, and RaFD.

BraTS The Multimodal Brain Tumor Segmentation Challenge (BraTS) 2018 (Menze et al., 2014; Bakas et al., 2017; 2018) provides a set of multi-institutional multimodal brain MRI scans with four modalities: a) native (T1) and b) post-contrast T1-weighted (T1Gd), c) T2-weighted (T2), and d) T2 Fluid Attenuated Inversion Recovery (T2-FLAIR). Following the setting in (Lee et al., 2019b), 218 and 28 subjects are randomly selected for training and test sets. The 2D slices at the same location are extracted from 3D MRI volumes for each of four modalities as one independent data sample in our experiments. In total, the training and testing datasets contain 40,148 and 5,340 images respectively. We resize the images of 240×240 to 256×256. Three labels are given for the brain tumor segmentation, i.e., enhancing tumor (ET), tumor core (TC), and whole tumor (WT).

ProstateX The ProstateX dataset (Litjens et al., 2014) contains multi-parametric prostate MR scans for 98 subjects. Here, we use the three modalities of each sample: 1) T2-weighted (T2), 2) Apparent Diffusion Coefficient (ADC), 3) high b-value DWI images (HighB). We randomly divide the dataset to 78 and 20 subjects for training and testing respectively. Similar to BraTS, 2D slices are extracted from 3D volumes for each modality. In total, the training and testing sets contain 3,540 and 840 images respectively. We resize the images of 384×384 to 256×256. Prostate regions are manually labeled as the whole prostate (WP) by board-certificated radiologists.

RaFD The Radboud Faces Database (RaFD) (Langner et al., 2010) contains eight facial expressions collected from 67 participants respectively: neutral, angry, contemptuous, disgusted, fearful, happy, sad, and surprised. We adopt images from three different camera angles (45°, 90°, 135°) along with three different gaze directions (left, frontal, right), 4,824 images in total. We treat the eight paired images in eight expression domains as an data sample. 54 participants (3,888 images) and 13 participants (936 images) are randomly divided for training and testing set. Following the setting in StarGAN (Choi et al., 2018), we crop the image with the face in the center and then resize them to 128×128.

3.1 Results of Multi-domain Image Completion

For comparison purpose, we firstly assume there are always only one domain missing for each sample and the algorithm will be evaluated for multiple times when one domain is missed at each

(a) BraTS

Methods	T1	T1Gd	T2	FLAIR
	NRMSE / SSIM	NRMSE / SSIM	NRMSE / SSIM	NRMSE / SSIM
MUNIT	0.4054 / 0.8904	0.3469 / 0.9084	0.4185 / 0.8702	0.4563 / 0.8728
StarGAN	0.3028 / 0.9346	0.2795 / 0.9351	0.5137 / 0.8473	0.4417 / 0.8931
CollaGAN	0.4729 / 0.8951	0.4952 / 0.8689	0.5243 / 0.8890	0.4299 / 0.8616
ReMIC w/o Recon	0.3502 / 0.9328	0.2349 / 0.9448	0.4485 / 0.8681	0.4214 / 0.8810
ReMIC	**0.1939** / 0.9629	0.2143 / 0.9546	0.2410 / 0.9469	0.2639 / 0.9369
ReMIC+Multi-Sample	0.2031 / **0.9637**	**0.2096 / 0.9563**	**0.2369 / 0.9490**	**0.2348 / 0.9395**

(b) ProstateX

Methods	T2	ADC	HighB
	NRMSE / SSIM	NRMSE / SSIM	NRMSE / SSIM
MUNIT	0.6904 / 0.4428	0.9208 / 0.4297	0.9325 / 0.5383
StarGAN	0.6638 / 0.4229	0.9157 / 0.3665	0.9188 / 0.4350
CollaGAN	0.8070 / 0.2667	0.7621 / 0.4875	0.7722 / 0.6824
ReMIC w/o Recon	0.8567 / 0.3330	0.7289 / 0.5377	0.8469 / 0.7818
ReMIC	0.4908 / 0.5427	0.2179 / 0.9232	**0.3894 / 0.9150**
ReMIC+Multi-Sample	**0.4742 / 0.5493**	**0.2171 / 0.9263**	0.3945 / 0.9116

Table 1: Multi-domain Medical Image Completion Results

time. Then we investigate a more general scenario when there are more than one missing domains and show that our proposed method is capable to handle the general random n-to-n image completion. Multiple quantitative metrics are used to evaluate the similarity between the generated image and original one, i.e., normalized root mean-squared error (NRMSE) and mean structural similarity index (SSIM) between generated images and target images. We compare our results with previous arts on all three datasets for this task. The results of the proposed method ("ReMIC"), ReMIC without reconstruction loss ("ReMIC w/o Recon") and ReMIC with cross-domain translation ("ReMIC+Multi-Sample", more details in Appendix C.2) are reported. Note that in our method, by leveraging the unified content code and sampling the style code for each domain respectively, the proposed model could handle any number of domain missing, which is more general and flexible for the random n-to-n image completion. Besides, these compared methods have their own limits.

MUNIT (Huang et al., 2018) conducts 1-to-1 image translation between two domains through representational disentanglement, as shown in Fig. 1(a). In experiments, we train and test MUNIT models between any pair of two domains. Without loss of generality, we select "neural" images to generate all the other domains by following the StarGAN setting, and "angry" image is used to generate "neural" one. In BraTS , "T1" is selected to generate other domains since it is the most common modality while "T1" is generated from "T1Gd". Similarly, "T2" is selected to generate other domains in ProstateX while "T2" is generated from "ADC".

StarGAN (Choi et al., 2018) adopts a mask vector to generate image in the specified target domain. Thus, different target domains could be generated from one source domain in multiple inference passes while only a single model is trained. This is actually a 1-to-n image translation, since only one domain can be used as input in StarGAN, we use the same domain pair match as MUNIT, which is also the same as the setting in the StarGAN (Choi et al., 2018).

CollaGAN (Lee et al., 2019a;b) carries out the n-to-1 image translation as shown in Fig. 1(b), where multiple source domains collaboratively generate the only one target domain which is assumed missing in inputs. But CollaGAN cannot deal with multiple missing domains. In CollaGAN experiments, we use the same domain generation setting as ours, that is, all the existing domains are used to complete the one missing domain in sequence.

Results of Medical Image Completion: Fig. 3 and Fig. 5 in appendix show some sample results of image completion on BraTS and ProstateX, and corresponding quantitative results are in Table 1. Each row shows the target and generated images of one domain with the assumption of that domain is missing in inputs. In comparison, our model achieves higher similarity to the target and also produce qualitatively more accurate images, e.g., a more accurate outstanding tumor region in BraTS and prostate regions are well-preserved in ProstateX. This is achieved by learning a better content code through factorized latent space in our method, which is essential in preserving the anatomical structures in medical images. Extended quantitative evaluation metrics are in Appendix.

Results of Facial Expression Image Translation: Fig. 4 shows a sample result of facial expression image completion on RaFD dataset. In each column, we show the target and generated images in

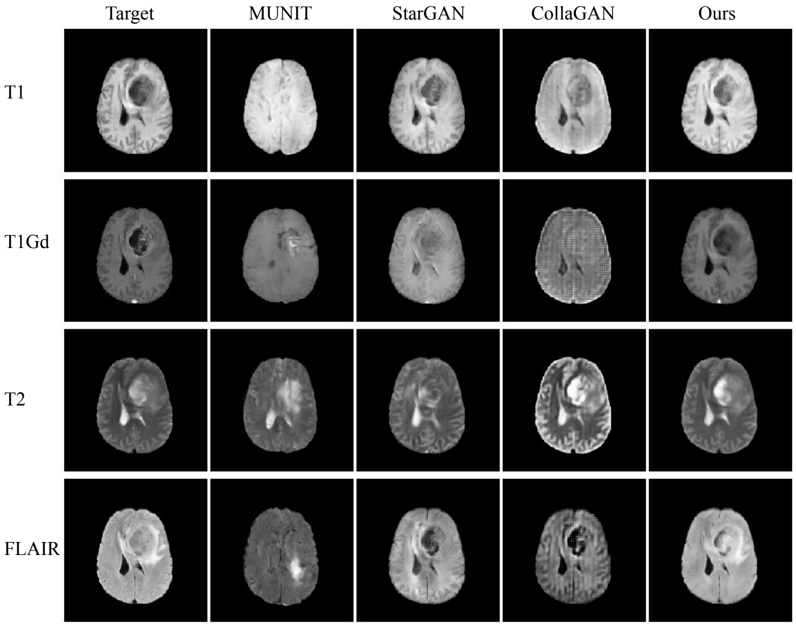

Figure 3: BraTS image completion results. Rows: 4 modalities. Columns: compared methods.

Methods	Neutral	Angry	Contemptuous	Disgusted
	NRMSE / SSIM	NRMSE / SSIM	NRMSE / SSIM	NRMSE / SSIM
MUNIT	0.1589 / 0.8177	0.1637 / 0.8156	0.1518 / 0.8319	0.1563 / 0.8114
StarGAN	0.1726 / 0.8206	0.1722 / 0.8245	0.1459 / 0.8506	0.1556 / 0.8243
CollaGAN	0.1867 / 0.7934	0.1761 / 0.7736	0.1856 / 0.7928	0.1823 / 0.7812
ReMIC w/o Recon	**0.1215** / 0.8776	0.1335 / 0.8556	**0.1192 / 0.8740**	0.1206 / 0.8559
ReMIC	0.1225 / **0.8794**	**0.1290 / 0.8598**	0.1217 / 0.8725	**0.1177 / 0.8668**
Methods	Fearful	Happy	Sad	Surprised
	NRMSE / SSIM	NRMSE / SSIM	NRMSE / SSIM	NRMSE / SSIM
MUNIT	0.1714 / 0.7792	0.1623 / 0.8073	0.1677 / 0.7998	0.1694 / 0.7884
StarGAN	0.1685 / 0.7943	0.1522 / 0.8288	0.1620 / 0.8227	0.1634 / 0.7974
CollaGAN	0.1907 / 0.7442	0.1829 / 0.7601	0.1783 / 0.7766	0.1888 / 0.7495
ReMIC w/o Recon	0.1321 / 0.8384	0.1399 / 0.8332	**0.1284 / 0.8597**	0.1333 / 0.8347
ReMIC	**0.1316 / 0.8395**	**0.1383 / 0.8406**	0.1301 / 0.8581	**0.1276 / 0.8484**

Table 2: RaFD multi-expression translation results.

one domain (expression), where we assume the target domain is missing in the inputs and need to be generated from one or more available domains. In this way, we evaluate the missing domains one by one sequentially. Compared with MUNIT and StarGAN results, our method could generate missing images with a better quality, especially in the generating details like teeth, mouth and eyes. This is paritally due to the fact that our method can incorporate complementary information from multiple domains, while MUNIT and StarGAN can only accept one domain as input source. For example, in the generation of "happy" and "disgusted" expressions, either MUNIT or StarGAN could not generate a good teeth and mouth region, since their source domain "neutral" does not contain the teeth. Compared with CollaGAN, our method could generate images with a better content through explicit disentangled representation learning in feature level instead of implicit cycle-consistency constraints in pixel level. The superior performance could also be observed in the NRMSE and SSIM value across all testing samples in Table 2 with all eight expressions.

3.2 Results of Image Segmentation with Missing Domain

Based on the missing-domain image completion as above, we show that our proposed method could go beyond image translation to solve the missing-domain image segmentation problem. Specifically, our model learns factorized representations by disentangling latent space, which could be efficiently leveraged for high-level tasks. As shown in Fig. 2, a segmentation branch is added after the learned

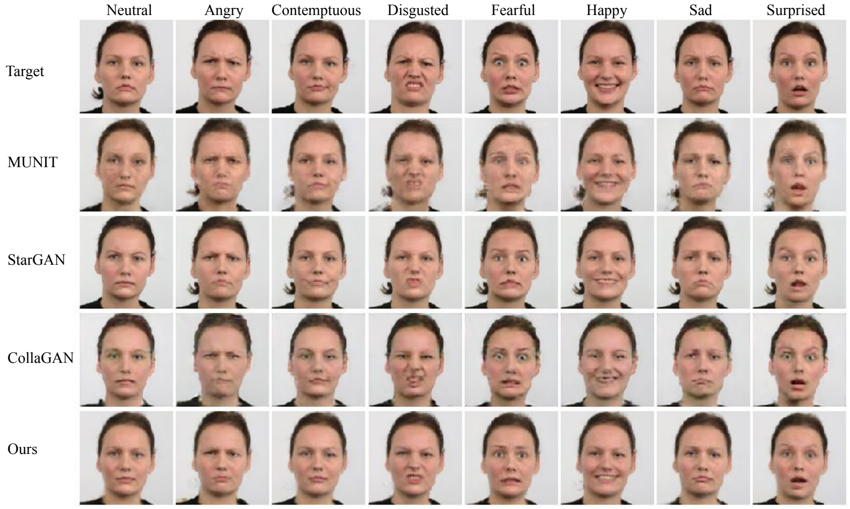

Figure 4: RaFD generation results. Columns: 8 expressions. Rows: compared methods.

Methods	BraTS				ProstateX		
	T1	T1Gd	T2	FLAIR	T2	ADC	HighB
Oracle+All	0.822				0.908		
Oracel+Zero	0.651	0.473	0.707	0.454	0.528	0.243	0.775
Oracle+Average	0.763	0.596	0.756	0.671	0.221	0.692	0.685
Oracle+NN	0.769	0.540	0.724	0.606	0.759	0.850	0.854
Oracle+MUNIT	0.783	0.537	0.782	0.492	0.783	0.708	0.858
Oracle+StarGAN	0.799	0.553	0.746	0.613	0.632	0.653	0.832
Oracle+CollaGAN	0.753	0.564	0.798	0.674	0.472	0.760	0.842
Oracle+ReMIC	0.789	0.655	0.805	0.765	0.871	0.898	0.891
ReMIC+Seg	0.806	0.674	0.822	0.771	**0.872**	**0.909**	**0.905**
ReMIC+Joint	**0.828**	**0.693**	**0.828**	**0.791**	0.867	0.904	0.904

Table 3: Missing-domain segmentation: Dice scores are reported here.

content codes to generate segmentation prediction mask. We evaluate the segmentation accuracy with Dice coefficient on both BraTS and ProstateX datasets.

The results of unified image completion and segmentation model as shown in Table 8 achieve the best dice score in both BraTS and ProstateX datasets. We train a fully supervised 2D U-shaped segmentation network (a U-net variation, Ronneberger et al. (2015)) without missing images as the “Oracle”. “Oracle+X” means that the results are computed by testing the missing images generated from the “X” method with the pretrained “Oracle” model. “All” represents the full testing set without any missing domains. “ReMIC+Seg” stands for using separate content encoders for generation and segmentation tasks in our proposed unified framework, while “ReMIC+Joint” indicates sharing the weights of content encoder for the two tasks. All methods obtain similar segmentation performances in ProstateX, but ReMIC is still relative closer to the Oracle results. In a more difficult segmentation task (like ET segmentation in BraTS), our proposed method shows significant improvement over other compared methods and even superior to the Oracle in some cases.

4 Conclusion

In this work, we propose a general framework for multi-domain image completion, given that one or more input images are missing. We learn shared content and domain-specific style encodings across multiple domains via the representational disentanglement. We also design several loss functions for accurate image generation, including image consistency loss, latent consistency loss, adversarial loss, and reconstruction loss. Our framework is flexible and can be easily extended to a unified generation and segmentation network. Extensive experiments validate the proposed method surpasses baseline methods and previous state-of-the-art methods on both multi-domain image generation and segmentation with missing domains.

References

Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4: 170117, 2017.

Spyridon Bakas, Mauricio Reyes, Andras Jakab, Stefan Bauer, Markus Rempfler, Alessandro Crimi, Russell Takeshi Shinohara, Christoph Berger, Sung Min Ha, Martin Rozycki, et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. *arXiv preprint arXiv:1811.02629*, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8789–8797, 2018.

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*, pp. 2672–2680, 2014.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1501–1510, 2017.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 172–189, 2018.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1857–1865. JMLR. org, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and emotion*, 24(8):1377–1388, 2010.

Dongwook Lee, Junyoung Kim, Won-Jin Moon, and Jong Chul Ye. Collagan: Collaborative gan for missing image data imputation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2487–2496, 2019a.

Dongwook Lee, Won-Jin Moon, and Jong Chul Ye. Which contrast does matter? towards a deep understanding of mr contrast using collaborative gan. *arXiv preprint arXiv:1905.04105*, 2019b.

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.

Geert Litjens, Oscar Debats, Jelle Barentsz, Nico Karssemeijer, and Henkjan Huisman. Computer-aided detection of prostate cancer in mri. *IEEE transactions on medical imaging*, 33(5):1083–1092, 2014.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pp. 700–708, 2017.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, 2017.

Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34 (10):1993–2024, 2014.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *International Workshop on Machine Learning in Medical Imaging*, pp. 379–387. Springer, 2017.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6924–6932, 2017.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, 2018.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Radialgan: Leveraging multiple datasets to improve target-specific predictive models using generative adversarial networks. *arXiv preprint arXiv:1802.06403*, 2018.

Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pp. 1116–1124, 2015.

Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2138–2147, 2019.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017a.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, 2017b.

A Extended Results for Multi-domain Image Completion

A.1 Qualitative Results of Image Completion for ProstateX

Due to the page limit, we put the quantitative results of image completion for ProstateX dataset here. As shown in Fig. 5, our method could generate images much closer to target images in all the three domains compared with other methods.

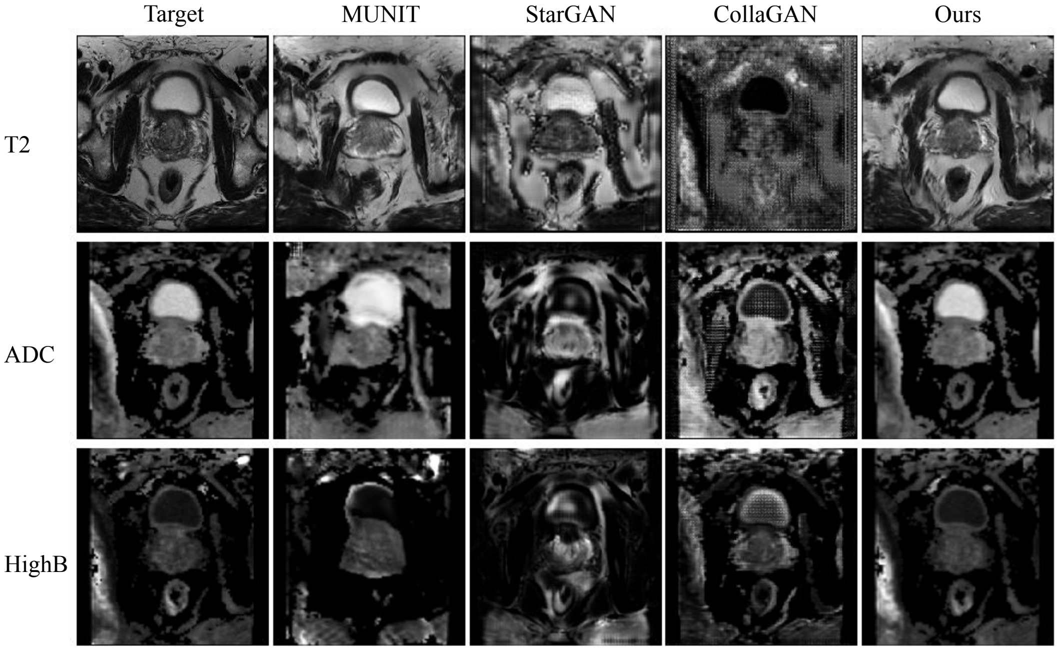

Figure 5: ProstateX generation results. Rows: 1) T2, 2) ADC, 3) HighB. Columns: 1) Ground truth, 2) MUNIT, 3) StarGAN, 4) CollaGAN, 5) ReMIC

A.2 Quantitative Results of Image Completion for Three Datasets

We demonstrate the quantitative results for multi-domain image completion with two more evaluation metrics: Mean Absolute Error (MAE), and Peak Signal-to-Noise Ratio (PSNR) for all three datasets. As shown in this section, Table 4, Table 5, and Table 6 are the extended full tables for Table 1 and Table 2 in the main text. The best results are in bold for each domain. It is shown that our method is able to generate images not only closer to the target images (lower MAE, NRMSE, and higher SSIM), but also with higher image quality (higher PSNR), from the results of three datasets.

In addition, for BraTS and ProstateX data, the generated images in the jointly trained generation-segmentation model are evaluated using the same metrics as shown in Table 4 and Table 5. The results indicate that adding segmentation branch does not bring an obvious benefit for image generation. This is because the segmentation sub-module mainly focuses on the tumor region which takes up only a small part among the whole slice image. Besides, we use dice loss as the segmentation training objective which might not be consistent with the metrics used to evaluate generated images that emphasize the pixel-level similarity.

B Extended Results for Image Segmentation with Missing Domain

Based on the missing-domain image completion, we show that our proposed method could go beyond image translation to solve the missing-domain segmentation problem. Specifically, our model learns factorized representations by disentangling latent space, which could be efficiently leveraged

Methods	T1	T1Gd
	MAE / NRMSE / PSNR / SSIM	MAE / NRMSE / PSNR / SSIM
MUNIT	0.0343 / 0.4054 / 22.2984 / 0.8904	0.0204 / 0.3469 / 25.6280 / 0.9084
StarGAN	0.0286 / 0.3028 / 24.7236 / 0.9346	0.0180 / 0.2795 / 27.4234 / 0.9351
CollaGAN	0.0383 / 0.4729 / 21.3504 / 0.8951	0.0304 / 0.4952 / 22.8241 / 0.8689
ReMIC w/o Recon	0.0302 / 0.3502 / 24.1283 / 0.9328	0.0145 / 0.2349 / 29.0389 / 0.9448
ReMIC	0.0181 / 0.1939 / **29.0687** / 0.9629	0.0134 / 0.2143 / 29.9335 / 0.9546
ReMIC+Multi-Sample	0.0186 / 0.2031 / 28.8312 / **0.9637**	**0.0131 / 0.2096 / 30.1349 / 0.9563**
ReMIC+Seg	0.0185 / 0.2038 / 28.5042 / 0.9607	0.0136 / 0.2193 / 29.6170 / 0.9507
ReMIC+Joint	**0.0179 / 0.1930** / 28.8452 / 0.9610	0.0136 / 0.2240 / 29.4216 / 0.9504
Methods	T2	FLAIR
	MAE / NRMSE / PSNR / SSIM	MAE / NRMSE / PSNR / SSIM
MUNIT	0.0298 / 0.4185 / 22.6869 / 0.8702	0.0374 / 0.4563 / 21.5905 / 0.8728
StarGAN	0.0375 / 0.5137 / 21.1905 / 0.8473	0.0332 / 0.4417 / 22.7630 / 0.8931
CollaGAN	0.0372 / 0.5243 / 21.3118 / 0.8890	0.0334 / 0.4299 / 22.3037 / 0.8616
ReMIC w/o Recon	0.0332 / 0.4485 / 22.0745 / 0.8681	0.0345 / 0.4214 / 22.2554 / 0.8810
ReMIC	0.0186 / 0.2410 / 27.7468 / 0.9469	0.0211 / 0.2639 / 26.9189 / 0.9369
ReMIC+Multi-Sample	**0.0185 / 0.2369 / 27.9594 / 0.9490**	0.0188 / **0.2348 / 27.6469 / 0.9395**
ReMIC+Seg	0.0204 / 0.2634 / 26.9036 / 0.9421	0.0197 / 0.2440 / 27.2777 / 0.9356
ReMIC+Joint	**0.0185** / 0.2421 / 27.6881 / 0.9457	**0.0185** / 0.2368 / 27.5816 / 0.9361

Table 4: BraTS results: Multi-domain image completion (full table)

Methods	T2	ADC
	MAE / NRMSE / PSNR / SSIM	MAE / NRMSE / PSNR / SSIM
MUNIT	0.1207 / 0.6904 / 15.6308 / 0.4428	0.1385 / 0.9208 / 13.8983 / 0.4297
StarGAN	0.1231 / 0.6638 / 15.9468 / 0.4229	0.1413 / 0.9157 / 13.8014 / 0.3665
CollaGAN	0.1480 / 0.8070 / **20.2846** / 0.2667	0.1063 / 0.7621 / 21.4448 / 0.4875
ReMIC w/o Recon	0.1580 / 0.8567 / 13.6738 / 0.3330	0.1070 / 0.7289 / 15.7083 / 0.5377
ReMIC	0.0840 / 0.4908 / 18.6200 / 0.5427	0.0253 / 0.2179 / 26.6150 / 0.9232
ReMIC+Multi-Sample	0.0810 / **0.4742** / 18.8986 / **0.5493**	**0.0250 / 0.2171 / 26.7024 / 0.9263**
ReMIC+Seg	**0.0871** / 0.5024 / 18.4236 / 0.5336	0.0272 / 0.2322 / 26.0828 / 0.9107
ReMIC+Joint	0.0881 / 0.5071 / 18.3206 / 0.5353	0.0288 / 0.2403 / 25.8024 / 0.9064
Methods	HighB	
	MAE / NRMSE / PSNR / SSIM	
MUNIT	0.0788 / 0.9325 / 16.9616 / 0.5383	
StarGAN	0.0883 / 0.9188 / 17.1168 / 0.4350	
CollaGAN	0.0571 / 0.7722 / 24.6687 / 0.6824	
ReMIC w/o Recon	0.0584 / 0.8469 / 17.8987 / 0.7818	
ReMIC	**0.0254 / 0.3894** / 24.7927 / **0.9150**	
ReMIC+Multi-Sample	0.0268 / 0.3945 / **24.8066** / 0.9116	
ReMIC+Seg	0.0272 / 0.4110 / 24.3277 / 0.9061	
ReMIC+Joint	0.0286 / 0.4359 / 23.8270 / 0.9006	

Table 5: ProstateX results: multi-domain image completion (full table)

for high-level tasks. As shown in Fig. 2, a segmentation branch is added after the learned content code to generate segmentation prediction map. We adopt dice loss as the segmentation loss in the training. We run the segmentation experiments on both BraTS and ProstateX datasets, and use dice score as evaluation metric. In the following, we look into two specific settings in missing-domain segmentation.

B.1 Missing-domain Imputation for Pretrained Segmentation Model

Suppose we have trained a segmentation model on a complete dataset with all images in available domains. Then during inference, this pretrained model will be used to predict segmentation results for new samples. For new subjects, some domains are missing. Straightforward solutions to complete the missing domains include zero filling, average image computed from all existing domains,

Methods	Neutral	Angry
	MAE / NRMSE / PSNR / SSIM	MAE / NRMSE / PSNR / SSIM
MUNIT	0.0547 / 0.1589 / 19.8469 / 0.8177	0.0556 / 0.1637 / 19.7303 / 0.8156
StarGAN	0.0545 / 0.1726 / 19.2725 / 0.8206	0.0543 / 0.1722 / 19.4336 / 0.8245
CollaGAN	0.0867 / 0.1867 / 24.3897 / 0.7934	0.0784 / 0.1761 / 24.8884 / 0.7736
ReMIC w/o Recon	0.0419 / **0.1215** / **22.2963** / 0.8776	0.0450 / 0.1335 / 21.4615 / 0.8556
ReMIC	**0.0399** / 0.1225 / 22.2679 / **0.8794**	**0.0431** / **0.1290** / **21.7570** / **0.8598**
Methods	Contemptuous	Disgusted
	MAE / NRMSE / PSNR / SSIM	MAE / NRMSE / PSNR / SSIM
MUNIT	0.0524 / 0.1518 / 20.2793 / 0.8319	0.0537 / 0.1563 / 19.9362 / 0.8114
StarGAN	0.0462 / 0.1459 / 20.7605 / 0.8506	0.0512 / 0.1556 / 20.0036 / 0.8243
CollaGAN	0.0883 / 0.1856 / 24.4246 / 0.7928	0.0869 / 0.1823 / 24.5366 / 0.7812
ReMIC w/o Recon	0.0402 / **0.1192** / **22.4073** / **0.8740**	0.0422 / 0.1206 / 22.1819 / 0.8559
ReMIC	**0.0401** / 0.1217 / 22.2414 / 0.8725	**0.0396** / **0.1177** / **22.4135** / **0.8668**
Methods	Fearful	Happy
	MAE / NRMSE / PSNR / SSIM	MAE / NRMSE / PSNR / SSIM
MUNIT	0.0605 / 0.1714 / 19.1714 / 0.7792	0.0571 / 0.1623 / 19.7709 / 0.8073
StarGAN	0.0552 / 0.1685 / 19.3516 / 0.7943	0.0504 / 0.1522 / 20.4397 / 0.8288
CollaGAN	0.0881 / 0.1907 / 24.1724 / 0.7442	0.0812 / 0.1829 / 24.5709 / 0.7601
ReMIC w/o Recon	0.0461 / 0.1321 / 21.4604 / 0.8384	0.0493 / 0.1399 / 20.9334 / 0.8332
ReMIC	**0.0455** / **0.1316** / **21.5295** / **0.8395**	**0.0469** / **0.1383** / **21.0465** / **0.8406**
Methods	Sad	Surprised
	MAE / NRMSE / PSNR / SSIM	MAE / NRMSE / PSNR / SSIM
MUNIT	0.0575 / 0.1677 / 19.3867 / 0.7998	0.0575 / 0.1677 / 19.3867 / 0.7998
StarGAN	0.0530 / 0.1620 / 19.7368 / 0.8227	0.0558 / 0.1634 / 19.6744 / 0.7974
CollaGAN	0.0783 / 0.1783 / 24.7656 / 0.7766	0.0856 / 0.1888 / 24.2375 / 0.7495
ReMIC w/o Recon	0.0450 / **0.1284** / **21.7430** / **0.8597**	0.0488 / 0.1333 / 21.3782 / 0.8347
ReMIC	**0.0436** / 0.1301 / 21.6384 / 0.8581	**0.0447** / **0.1276** / **21.7793** / **0.8484**

Table 6: RaFD results: Multi-domain image completion (full table)

and the nearest neighbor (NN) among training samples. We show the dice scores for these baseline methods in Table 7. Oracle results give the average testing dice score when all the domains are available in inference. Each column shows the dice score of segmentation prediction when the current domain is missing during inference. Moreover, based on image translation methods, we can generate fake images for missing domain imputation, and the results for different methods are also shown in Table 7. We show that our proposed method achieves the best dice score compared with all aforementioned baselines and other GAN-based image translation methods. This also indicates the our method could generate superior images preserving a better content representation through disentangled latent space. Furthermore, from the results in Table 7, we know that the T1Gd modality and the T2 modality are the most significant contrasts in the segmentation of BraTS and ProstateX data, whose missing will cause a severe decrease in dice score performance. Our method could alleviate such a loss to a large extent. Here, the dice score for BraTS is the average number for the three classes: WT, TC, and ET. Please see Table 9 in Appendix for a full table with all per-class dice scores.

B.2 Missing-domain Segmentation Training

Suppose we would like to train a segmentation model for a new data set, but most patients in this cohort just contain a random subset of all required domains. In this scenario, it is definitely not efficient to just use the most common domain overlapped by most patients. One simple solution is to complete all the missing images in training set by some imputation method, such as zero-filling image, average image, or generating images via image translation model. The results for these methods are shown in Table 8. More advanced, based on the content code learned in our model, we could develop a join model for multi-task learning of both generation and segmentation. Specifically, another branch of segmentation generator is added after content encoder to generate segmentation map. By optimizing the generation loss in Eq. 6 and segmentation loss in Eq. 2.2 simultaneously, the model could adaptively learn how to generate missing images to promote segmentation performance.

The results of jointly learned model as shown in Table 8 achieve the best dice score in both BraTS and ProstateX datasets. "ReMIC+Seg" stands for using separate content encoders for generation and segmentation tasks, while "ReMIC+Joint" indicates sharing the weights of content encoder for the two tasks. Note that the baseline methods get better results after retraining the model on the missing data, since the model is trained to fit to the exact input format by optimizing the segmentation objective under the supervision of segmentation labels. However, our method can still get the best results through adaptive learning model. In ProstateX data, the segmentation of whole prostate region is not very challenging and the numbers among different methods do not differ a lot. But in more difficult segmentation tasks like ET segmentation in BraTS, our proposed method shows an apparent advantage over other methods as shown in Table 10.

Methods	BraTS				ProstateX		
	T1	T1Gd	T2	FLAIR	T2	ADC	HighB
Oracle	0.822				0.908		
Zero	0.651	0.473	0.707	0.454	0.528	0.243	0.775
Average	0.763	0.596	0.756	0.671	0.221	0.692	0.685
NN	0.769	0.540	0.724	0.606	0.759	0.850	0.854
MUNIT	0.783	0.537	0.782	0.492	0.783	0.708	0.858
StarGAN	0.799	0.553	0.746	0.613	0.632	0.653	0.832
CollaGAN	0.753	0.564	0.798	0.674	0.472	0.760	0.842
ReMIC	**0.819**	**0.641**	**0.823**	**0.784**	**0.863**	**0.907**	**0.903**

Table 7: Missing-domain imputation for pretrained segmentation model

Methods	BraTS				ProstateX		
	T1	T1Gd	T2	FLAIR	T2	ADC	HighB
Oracle	0.822				0.908		
Zero	0.811	0.656	0.823	0.775	0.868	0.899	0.897
Average	0.796	0.604	0.788	0.759	0.856	0.885	0.897
ReMIC	0.789	0.655	0.805	0.765	0.871	0.898	0.891
ReMIC+Seg	0.806	0.674	0.822	0.771	**0.872**	**0.909**	**0.905**
ReMIC+Joint	**0.828**	**0.693**	**0.828**	**0.791**	0.867	0.904	0.904

Table 8: Missing-domain segmentation

B.3 Extended Segmention Results for BraTS Dataset

Firstly, we shows full tables with the per-class dice scores for BraTS segmentation results in Table 9 and Table 10. Compared with WT and TC classes, ET class is definitely more challenging in the brain tumor segmentation task, since enhancing tumor always just occupy a very small region among the whole tumor. In the ET class, we can see our method outperforms the other methods to a large extent.

Furthermre, we validate our method can work not only for 2D image segmentation but also 3D image segmentation. When a 3D volumetric image is missing in some domain, we deploy our method to generate 2D images per slice and stack them to build the whole 3D volumetric image in this domain. As shown in Table 9, we evaluate the per-class dice score for missing-domain imputation with the oracle model trained from complete-domain 3D segmentation. The results show our method could give a better performance in most domains. During experiment, the smoothness among different slices in 3D image generation is a concern that could be improved.

C Ablative Study

C.1 Random Multi-domain Image Completion

To show the superiority of our method in dealing with a random subset of missing domain, we train our model with randomly missing domains in training samples for RaFD dataset. Among all

Methods		T1	T1Gd	T2	FLAIR
		WT / TC / ET	WT / TC / ET	WT / TC / ET	WT / TC / ET
2D	Oracle	0.910 / 0.849 / 0.708			
	Zero	0.771 / 0.609 / 0.572	0.872 / 0.539 / 0.008	0.755 / 0.690 / 0.677	0.458 / 0.468 / 0.435
	Average	0.870 / 0.744 / 0.674	0.882 / 0.603 / 0.303	0.849 / 0.732 / 0.686	0.655 / 0.710 / 0.648
	NN	0.883 / 0.765 / 0.660	0.871 / 0.564 / 0.186	0.811 / 0.720 / 0.642	0.534 / 0.669 / 0.614
	MUNIT	0.886 / 0.785 / 0.679	0.872 / 0.552 / 0.187	0.882 / 0.781 / 0.682	0.408 / 0.541 / 0.527
	StarGAN	0.897 / 0.795 / 0.704	0.886 / 0.588 / 0.184	0.851 / 0.725 / 0.661	0.570 / 0.664 / 0.604
	CollaGAN	0.860 / 0.747 / 0.651	0.864 / 0.576 / 0.252	0.882 / 0.811 / 0.700	0.663 / 0.697 / 0.663
	ReMIC	**0.909 / 0.834 / 0.714**	**0.899 / 0.669 / 0.354**	**0.905 / 0.855 / 0.709**	**0.853 / 0.807 / 0.691**
3D	Oracle	0.909 / 0.867 / 0.733			
	Zero	0.876 / 0.826 / 0.694	0.884 / 0.574 / 0.020	0.901 / 0.865 / 0.728	0.661 / 0.730 / 0.643
	Average	0.880 / 0.814 / 0.640	0.854 / **0.618 / 0.282**	0.838 / 0.801 / 0.695	0.713 / 0.732 / 0.675
	NN	0.890 / 0.829 / 0.703	0.859 / 0.538 / 0.081	0.790 / 0.799 / 0.704	0.472 / 0.686 / 0.607
	ReMIC	**0.905 / 0.864 / 0.722**	**0.888** / 0.614 / 0.273	**0.902 / 0.871 / 0.734**	**0.855 / 0.850 / 0.724**

Table 9: BraTS results: Missing-domain imputation for pretrained segmentation model (full tabel)

Methods	T1	T1CE	T2	FLAIR
	WT / TC / ET	WT / TC / ET	WT / TC / ET	WT / TC / ET
Oracle	0.910 / 0.849 / 0.708			
Zero	0.904 / 0.818 / 0.710	0.888 / 0.687 / 0.394	0.907 / 0.842 / 0.720	0.841 / 0.793 / 0.692
Average	0.905 / 0.798 / 0.685	0.898 / 0.603 / 0.312	0.897 / 0.803 / 0.663	0.846 / 0.768 / 0.663
ReMIC	0.908 / 0.783 / 0.676	0.897 / 0.685 / 0.382	0.901 / 0.815 / 0.698	0.851 / 0.779 / 0.665
ReMIC+Seg	0.911 / 0.819 / 0.687	0.902 / 0.708 / 0.411	**0.910** / 0.839 / **0.716**	0.850 / 0.792 / 0.671
ReMIC+Joint	**0.915 / 0.859 / 0.710**	**0.910 / 0.740 / 0.430**	0.909 / **0.858 / 0.716**	**0.860 / 0.823 / 0.691**

Table 10: BraTS results: Missing-domain segmentation (full table)

the eight domains, we assume that each sample contains at least one existing domain randomly. In other words, there are possibly 0∼7 missing domains for each training sample. This is a very challenging setting where the number and the choice of missing domains are totally random. This difficult problem cannot be solved by existing works, and to our best knowledge, our proposed method is the first one that could achieve the random n-to-n image translation. In testing, we evaluate the model with different number of input domains. We show the results of three testing samples shown in Figs. 6∼ 8. The top half of each column shows the input domain(s), where the missing domains are filled with zeros. The bottom half of each column shows the generation results for all the eight domains no matter if it exists in input domains or not. Firstly, in our method, no matter how many and which domains are available, the model could generate images for all the domains including missing ones in an one-time inference. Comparing the results in each row, we could see that the domain-specific style and domain-shared content are all preserved well even when we push the limit of existing domain(s) to be only one. In addition, when the number of visible domains increase, the content in each domain image is enhanced gradually and gets closer to the target image. This illustrates that our model is efficiently learning a better content code through representation disentanglement as the source domains provide more complementary information.

C.2 Multi-sample Learning

Based on the proposed model in Fig. 2, we further propose a more advanced model and training strategy when multiple samples are inputted at one time. Generally, based on the assumption of partially shared latent space, we assume that the factorized latent code can independently represent the corresponding content and style information in the input image. Then by exchanging the content codes from two independent samples, it should be able to reconstruct the original input image by recombining the style code extracted from the other sample. Based on this idea, we build a comprehensive model with cross-domain training between two samples. Similarly as the framework in Fig. 2, the image and code consistency loss and image reconstruction loss are also constrained through the encoding and decoding procedure. The results of multi-sample learning are shown in Table 1 named as "ReMIC+Multi-Sample".

C.3 Analysis of missing-domain segmentation results

To better understand why our method is a better solution in missing-domain imputation for multi-domain recognition tasks like segmentation, we demonstrate three randomly selected testing samples in BraTS dataset as shown in Fig. 9. Rows 1$\sim$3 shows the first sample, and the left two samples are shown in the same format. In the first sample, the first row shows real images in four domains and its ground truth segmentation labels. When some domain for this sample is randomly missing, a straightforward solution is to search through all the available training data and find the nearest neighbor (NN) to complete the missing image. We search the nearest neighbor according to the Euclidean distance in 2D image space. The second row shows the images in four domains and the segmentation map for the NN sample, which actually looks very similar to the target sample visually. However, we note that the tumor region is seriously different between the target sample and its NN sample, which shows the NN image is not a good missing imputation in semantics. To cope with this issue, our proposed method could generate images for missing domains with not only pixel-level similarity but also similar tumor regions, which are the most significant parts in tumor segmentation task. As shown in the third row, the generated images in four domains are very close to the target images. The segmentation map shows the prediction results when the generated T1 image is used as imputation in inputs, which gives a segmentation mask very close to the ground truth. These results illustrate the superiority of our method, which results from the disentangled representation learning in feature level.

D Implementation Details

Here, we describe the implementation details of our method. We will open source all the source codes and models if get accepted.

D.1 Hyperparameters

In our algorithm, we use the Adam optimizer (Kingma & Ba, 2014) with $\beta_1 = 0.5, \beta_2 = 0.999$. The learning rate is 0.0001. We set the loss weights in Equation 6 as $\lambda_{adv} = 1, \lambda_{cyc}^{x} = 10, \lambda_{cyc}^{c} = 1, \lambda_{cyc}^{s} = 1, \lambda_{recon} = 20$. For comparison purpose, we train the model to 150,000 / 100,000 / 100,000 iterations for BraTS, ProstateX, and RaFD datasets respectively, and compare the results across MUNIT, StarGAN, CollaGAN, and ours ReMIC in all datasets. In ReMIC, we set the dimension of the style code as 8 for comparison purpose with MUNIT. For image generation during testing, we use a fixed style code of 0.5 in each dimension for both MUNIT and ReMIC to compute quantitative results.

D.2 Network Architectures

The network structures of ReMIC is developed on the backbone of MUNIT model. We describe the details of each module here.

Unified Content Encoder: consists of a down-sampling module and residual blocks to extract contexture knowledge from all available domain images in inputs. The down-sampling module contains a 7 × 7 convolutional block with stride 1 and 64 filters, and two 4 × 4 convolutional blocks with stride 2 and, 128 and 256 filters respectively. The strided convolutional layers downsample the input to features maps of size $W/4 \times H/4 \times 256$, where W and H are the width and height of input image. Next, there are four residual blocks, each of which contains two 3 × 3 convolutional blocks with 256 filters and stride 1. We apply Instance Normalization (IN) (Ulyanov et al., 2017) after all the convolutional layers. Note that the proposed unified content encoder accepts images of all domains as input (missing domains are filled with zeros in initialization), and learns a universe content code complementarily and collaboratively, which are different from MUNIT.

Style Encoder: contains a similar down-sampling module and several residual blocks, which is followed by a global average pooling layer and a fully connected layer to learn the style vector. The down-sampling module is developped using the same structure as that in the unified content encoder above, and two more 4 × 4 convolutional blocks with stride 2 and 256 filters are followed. The final fully connected layer generates style code as a 8-dim vector. There is no IN applied to the style encoders to keep the original feature means and variances with style information (Huang & Belongie, 2017).

Generator: includes four residual blocks, each of which contains two 3×3 convolutional blocks with 256 filters and stride 1. Two nearest-neighbor upsampling layers and a 5×5 convolutional block with stride 1 and, 128 and 64 filters respectively are followed to up-sample content codes back to the original image size. Finally, there is a a 7×7 convolutional block with stride 1 to output the reconstructed image. In order to incorporate the style code in the generation process, the Adaptive Instance Normalization (AdaIN) (Huang & Belongie, 2017) is applied to each residual block as follows (Huang et al., 2018):

$$\mathit{AdaIN}(z, \gamma, \beta) = \gamma\Big(\frac{z - \mu(z)}{\sigma(z)}\Big) + \beta \tag{8}$$

where z is the activation from the last convolutional layer. $\mu(z)$ and $\sigma(z)$ are the channel-wise mean and standard deviations of the activation. γ and β are the affine parameters in the AdaIN layers that are generated from style codes via a multi-layer perceptron (MLP). In this way, the input style code controls the generated style information through the affine transformation in the AdaIN layers in all generators (Huang & Belongie, 2017).

Discriminator: includes four 4×4 convolutional blocks with stride 2 and, 64, 128, 256, and 512 filters in sequence. The Leaky ReLU activation with slope 0.2 is applied after convolutional layers. A multi-scale discriminator (Wang et al., 2018) is used to consider the results at three scales together. In adversarial training, we adopt LSGAN objective (Mao et al., 2017) as adversarial loss to learn to generate realistic images.

Segmentor: We adopt a segmentation net with a U-Net shape (Ronneberger et al., 2015). In order to build a joint model with the image generation modules, we build a variant U-Net: the downsampling part share the same structure as the content encoder while the upsampling part have the same layers as the generator as described above. Similar as the original U-Net (Ronneberger et al., 2015), we also adopt the skip-connections between the downsampling and upsampling layers in the our segmentation model.

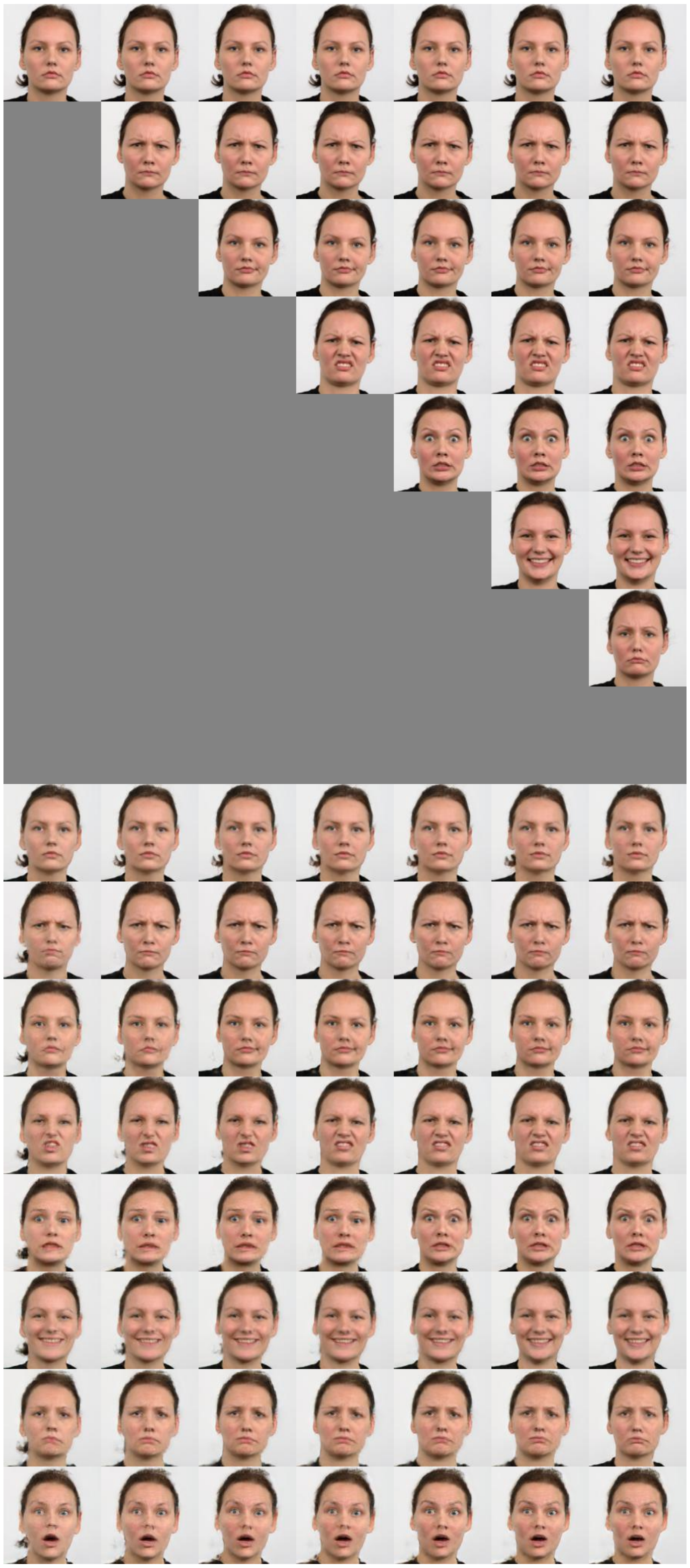

Figure 6: Random multi-domain image completion results. Rows 1-8 are input images, and rows 9-16 are generated images when different numbers of input images are given. Each column demonstrates the images in one domain in the order of “neutral”, “angry”, “contemptuous”, “disgusted”, “fearful”, “happy”, “sad”, “surprised”.

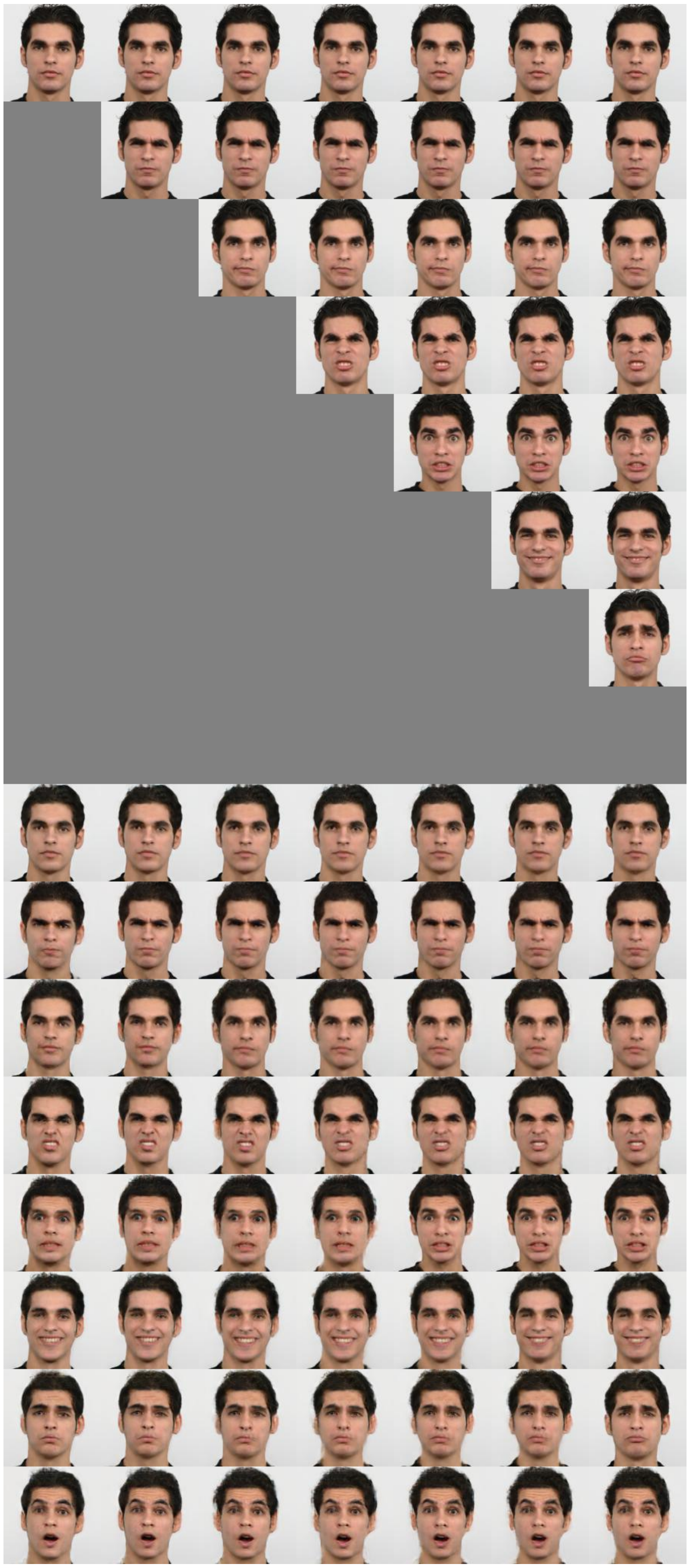

Figure 7: Random multi-domain image completion results. Rows 1-8 are input images, and rows 9-16 are generated images when different numbers of input images are given. Each column demonstrates the images in one domain in the order of "neutral", "angry", "contemptuous", "disgusted", "fearful", "happy", "sad", "surprised".

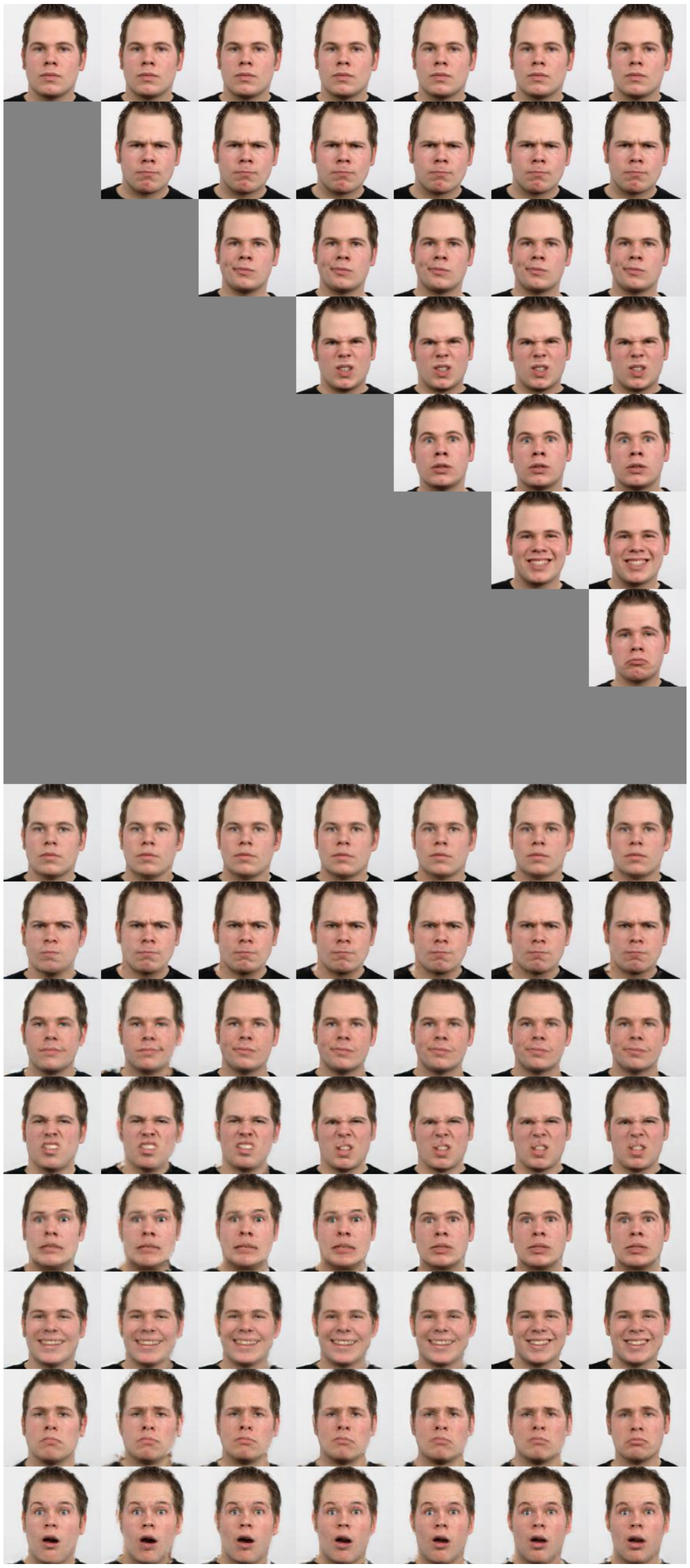

Figure 8: Random multi-domain image completion results. Rows 1-8 are input images, and rows 9-16 are generated images when different numbers of input images are given. Each column demonstrates the images in one domain in the order of “neutral”, “angry”, “contemptuous”, “disgusted”, “fearful”, “happy”, “sad”, “surprised”.

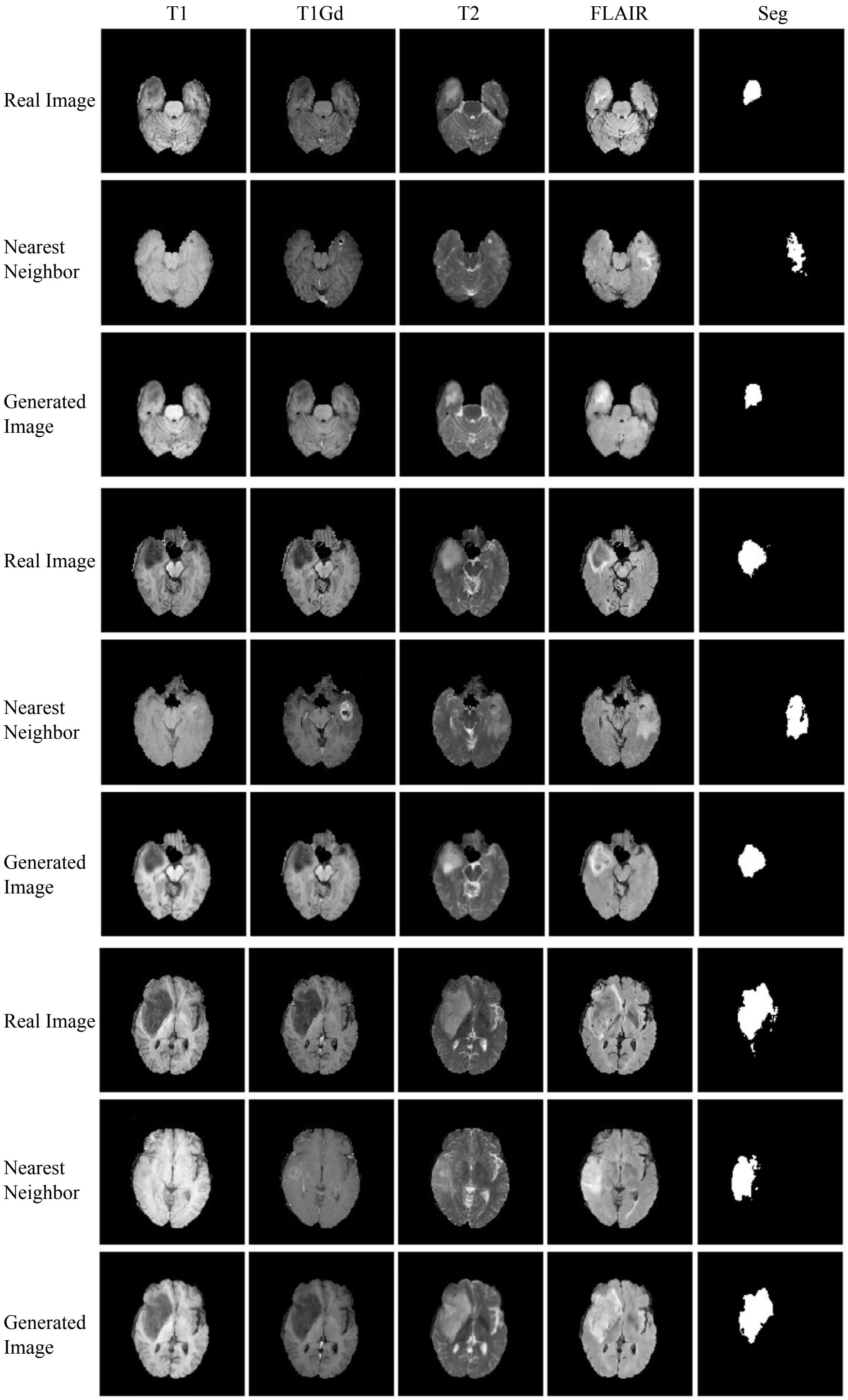

Figure 9: Missing-domain segmentation results of three testing samples in BraTS. Every three rows show results for one testing sample. For each testing sample, we show: 1) real images with groundtruth segmentation label, 2) nearest neighbor searched from training data with its segmentation label, 3) generated images using our method and segmentation prediction when T1 is missing.