Domain Translation via Latent Space Mapping

First Author[1 [0000−1111−2222−3333]], Second Author[2,3 [1111−2222−3333−4444]], and Third Author[3 [2222−−3333−4444−5555]]

No Institute Given

Abstract. In this paper, we investigate the problem of multi-domain translation : given an element a of domain A, we would like to generate a corresponding b sample in another domain B, and vice versa. Acquiring supervision in multiple domains can be a tedious task, also we propose to learn this translation from one domain to another when supervision is available as a pair $(a, b) \sim A \times B$ and leveraging possible unpaired data when only $a \sim A$ or only $b \sim B$ is available. We introduce a new unified framework called Latent Space Mapping (LSM) that exploits the manifold assumption in order to learn, from each domain, a latent space. Unlike existing approaches, we propose to further regularize each latent space using available domains by learning each dependency between pairs of domains. We evaluate our approach in three tasks performing i) synthetic dataset with image translation, ii) real-world task of semantic segmentation for medical images, and iii) real-world task of facial landmark detection. Source code is publicly available[1].

Keywords: Domain translation · Semi-supervised learning · Unsupervised learning · Weakly-supervised learning.

1 Introduction

In many machine learning tasks, different modalities can be modeled as different domains and different data as different views of the same reality. For example considering an autonomous vehicle context, the RGB camera, the depth map, and the segmentation map can be considered as three views of the same reality. Often one domain can be considered as an input domain, e.g. a CT (computed tomography) scan of a patient, and other domains as output domain, e.g. organ segmentation of the scan. Here we do not consider an input nor output domain, but rather we would like to learn to translate from any domain to another one. This definition masks the classical definition of an input and output domain, as every domain can be a possible input or output. While the rising amount of available data has brought great results in domain translation tasks in a fully supervised fashion, in some fields the quantity of available supervision is limited and hard to obtain, as in the medical field where a domain expert is required to create a hand-made segmentation [21, 29], and this low supervision setting often reduces the performance of classical deep learning model.

[1] Links to code will be inserted in the final version.

Acquiring complete supervision to learn this translation across the different domains can be a tedious task, since for one view all the other corresponding views in the other domains are required. We now consider the setting of incomplete supervision, where for one view, the other view might or might not be available.

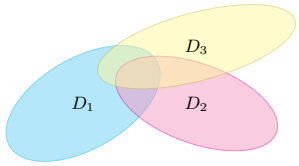


Fig. 1: Set of three domains D_1, D_2 and D_3, each views can have or not have a corresponding view in the other domain, creating a situation of semi-supervised learning

We investigate the task of multi-domain translation, i.e., given a set of n domains $\{D_1, \ldots, D_n\}$, we would like to learn a mapping function $\forall i, j \quad f_{i \to j} : D_i \to D_j$ allowing to translate a view from the domain D_i into its view in D_j. An illustration with $n = 3$ is given in Figure 1 where multiple possible configurations of the triple (d_1, d_2, d_3) are possible, and each point does not necessarily have associated view in the other two domains available (e.g. example of available data: (d_1, d_2), (d_2, d_3), or only (d_1)).

Many classical machine learning tasks fall into this domain translation framework, provided that one can reconstruct a view d_i given another view d_j. For example, image captioning and machine translation can be viewed as domain translation problems. In Image captioning, one domain is the image and the other is its description, and for machine translation both domains are textual. Previous approaches leverage useful assumptions and inductive bias, a common one in domain translation is the shared latent space assumption [4, 9, 11, 15, 28] - associated views of different domains can be embedded as the same code in a latent space.

In this paper, we propose LSM unified framework, which allows learning to translate one view from a domain to another domain in a semi-supervised setting when either all the views in each domain are available, or when an arbitrary number of views are available. We do so by learning a latent space for each of the domains where each view can be embedded in a manifold.

Each manifold is further constrained by the mean of distance and adversarial losses allowing LSM to learn a better representation by leveraging the intra-domain dependencies (dependencies within one domain, e.g. the value of one image pixel or segmentation mask regarding its surrounding pixel values) and inter-domain dependencies (the dependencies between multiple domains, e.g. the pixel of an image with regard to the associated pixel of the segmentation mask).

The rest of this paper is organized as follows, Section 2 review related work to domain translation, Section 3 present how LSM framework is structured and trained, in Section 4 we describe how domain translation can be applied to three tasks (semantic segmentation, image to image translation, and facial landmark detection). The detailed settings of the training are provided in the Appendix.

2 Related Works

Domain translation is related to other tasks like semi-supervised learning (SSL), and domain adaptation (DA). While SSL and DA differ from domain translation, these three tasks leverage common assumptions and often use similar methods. The goal of SSL is to learn from labeled and unlabeled samples, but in SSL an input domain and output domain are clearly identified. In domain translation, a sample can be instantiated in A but not in B, in B but not in A, or in A and B. None of the domains can be considered either as 'input' nor 'output'. In DA, an input domain and an output domain are also often considered, but the input domain contains two sub-domains: a source domain and a target domain, in this setting close to SSL, the source domain contains labeled examples and the target domain is often left without supervision. The discrepancy between the two inputs domains prevents from just merging them together. In this work, we leverage two useful SSL assumptions that we will present: the Manifold Assumption that we use to learn the intra-domain dependencies, and the Shared Latent Space Assumption that we relax in order to learn inter-domain dependencies. We will then present methods that exploit the same type of assumptions, and how we take a different interpretation of these assumptions.

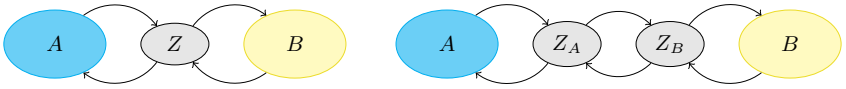
2.1 Manifold Assumption

According to the manifold assumption, high-dimensional data can be embedded in a lower-dimensional space [5], a latent space, with a lower degree of freedom considered. For the algorithm, working with this latent code representation of the data is easier than working with the original high-dimensional data, and learning this latent space helps to capture the underlying data structure - the intra-domain dependencies. Several architectures such as the auto-encoder and variational auto-encoder have been used to learn this latent space [3, 12, 13, 18]. While learning intra-domain dependencies has shown effectiveness, allowing to use additional data without associated views in the other domains [21], this approach alone does not leverage inter-domain dependencies when there is no supervision. The next assumption help to alleviate this issue.

2.2 Shared Latent Space Assumption

Deriving from the Manifold Assumption and mainly used in DA, the Shared Latent Space Assumption state that associated views of different domains can be mapped to a common latent code. If a view $d_i \sim D_i$ is associated with

view $d_j \sim D_j$, then $\tilde{d}_i = \tilde{d}_j$. Learning to align each manifold in the same shared structure allows learning the inter-domain dependency between views of different domains. But this every-to-one mapping can be seen as an over-simplification, as multiple modalities can have very dissimilar features (e.g. consider an image and its pixel-wise semantic segmentation mask, the mask does not contain in itself information about the texture), this inter-domain discrepancy coupled with the every-to-one mapping could hinder the model when learning the translation.



(a) A single shared latent space (b) Multiple latent spaces space

Fig. 2: Two type of latent space are commonly used in domain translation (a) a single shared latent space, (b) or using multiple latent space

2.3 Domain Translation

We will give an overview of common domain translation methods while classifying them according to the previously defined assumptions in order to better understand how we take different choice from them. These models can be divided into roughly two types illustrated in Figure 2.

Methods using a **single shared latent** space can use adversarial learning on the feature space [4, 9, 11, 15, 28], and possibly also on the original high-dimensional space [6, 11, 14, 17, 22–24, 37], using generative adversarial networks (GAN [10]) to generate a view from a latent code. GAN has been notoriously hard to train due to the min-max optimization that can lead to vanishing gradient [1] and is subject to mode collapse [33] (a mode collapse occurs when the generation is almost deterministic and the generator always create the same output). Building on top of these existing works, models with **multiple latent spaces** often use disentanglement learning, dividing the latent code into two-part: a domain agnostic content code, and a domain-specific style code. This more sophisticated setting allows learning a domain translation function that would combine a content code from a source domain and the style code of the targeted domain [16, 20, 22, 25, 32, 36]. Learning a disentangled representation is not straightforward as the definition in itself of content and style is ambiguous, these approaches bring great results at the cost of additional complexity and inductive bias (e.g. using adversarial learning between the content code and the style code).

2.4 Latent Space Mapping Assumption

We will now present how LSM uses the previously defined concept and how it differs from existing works by taking a different interpretation of the pre-

sented assumptions. LSM use the manifold assumption by, for each domain D_i, learning a latent space where a view d_i is embedded in its latent code $\tilde{d}_i$. This allows the approach to modelize intra-domain dependencies. Additionally, we learn the inter-domain dependencies using the shared latent space assumption, but here we decide to take a different approach from the classical full-shared latent space while keeping its simplicity. Instead of using a single shared latent space, we relax the assumption by supposing the existence of a function $L_{i\to j} : \tilde{D}_i \to \tilde{D}_j$ allowing to translate a latent code from one domain to the latent code of another domain. The shared assumption is then applied between translated codes $L_{i\to j}(\tilde{d}_i)$ and codes $\tilde{d}_j$ from the original domain. Similar ideas has been approached with success in existing works: in IODA [21] by the mean of pre-training an input encoder using input data, an output decoder using output data to initialize a network, and learning the final task in a supervised fashion. And in SOP [2] where there is no pre-training but the learning of this latent space is done in an unsupervised fashion as a regularization task. But SOP only uses the manifold assumption and does not take into account the discrepancy between a translated latent code and a latent code from the original domain. To alleviate this issue, we will use a framework similar to that of the SOP. We further regularizing each latent space by making the assumption that $L_{i\to j}(\tilde{d}_i)$ should be close to $\tilde{d}_j$.

3 LSM Framework

We now present LSM framework, illustrated in Figure 3, for domain translation between multiple modalities and multiple domains, by learning functions $f_{D_i\to D_j} : D_i \to D_j$. The framework learn intra-domain dependencies, for each view $d_i \sim D_i$, the model produces its embedding $\tilde{d}_i$, this can been done in an unsupervised way by the mean of reconstruction task. It also relaxes the shared latent space assumption by supposing the existence of a function $L_{i\to j}$ allowing to translate a latent code $\tilde{d}_i$ to another domain latent space $\tilde{d}_j$, the shared assumption is then applied after the translation by the mean of a distance loss when paired information is available or by the mean of adversarial in an unpaired setting.

For clarity concern and without loss of generality, we will now consider only two domains X and Y, and we will present the translation from domain X to domain Y, the same losses are applied when translating from Y to X. The under-script $x \to y$ will be used on functions that are specific to the translation of X to Y and has to be duplicated in the case of Y to X. Other functions are only applied once. This approach leverage three types of data configuration, either learning from view $x \sim p(x)$ when y is not available, from $y \sim p(y)$ when it is provided without corresponding x, or from paired views $(x, y) \sim p(x, y)$. Our goal is to learn a translation function $f_{x\to y} : X \to Y$ from domain X to domain Y as $\hat{y} = f_{x\to y}(x)$. Using the manifold assumption we introduce two latent spaces $\tilde{x}$ (resp. $\tilde{y}$) where the view x (resp. y) could be encoded and where the code would retain a summarized version of the original data. To learn this

latent space, two auto-encoders are used: $\tilde{x} = E_x(x)$ and $\tilde{y} = E_y(y)$ mapping a view to its latent code, and $\hat{x} = D_x(\tilde{x})$ and $\hat{y} = D_y(\tilde{y})$ mapping the latent code to its original view. An additional function performs the mapping between latent space X to Y as $\tilde{y} = L_{\tilde{x}\to\tilde{y}}(\tilde{x})$. Given these latent spaces and functions, $f_{x\to y}$ can be reformulated as: $\hat{y} = f_{x\to y}(x) = D_y(\tilde{y}) = D_y \circ L_{\tilde{x}\to\tilde{y}}(\tilde{x}) = D_y \circ L_{\tilde{x}\to\tilde{y}} \circ E_x(x)$.

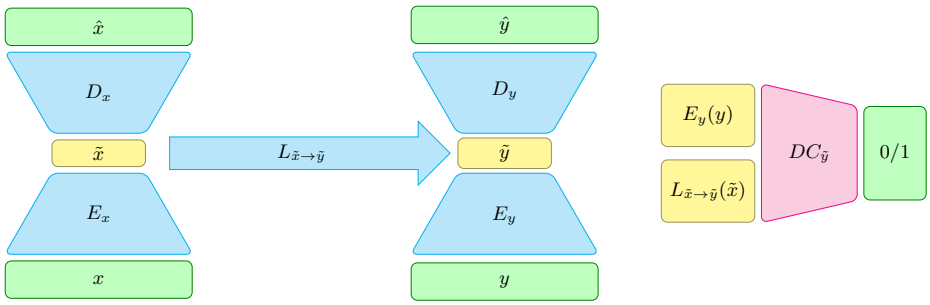


Fig. 3: Illustration of LSM framework. The translation is presented from the point of view of $X \to Y$

3.1 Learning X to Y Domain Translation

When a pair is available as $(x, y) \sim p(x, y)$, the model performs as a fully supervised method, predicting y from x.

$$\hat{y} = D_y \circ L_{\tilde{x}\to\tilde{y}} \circ E_x(x) \tag{1}$$

This allows learning the translation network by optimizing the supervised loss

$$\min_{\theta_{E_x}, \theta_{L_{\tilde{x}\to\tilde{y}}}, \theta_{D_y}} \mathbb{E}_{p(x,y)} \mathcal{L}_{x\to y}(D_y \circ L_{\tilde{x}\to\tilde{y}} \circ E_x(x), y) = \mathcal{L}_{x\to y}(\hat{y}, y) \tag{2}$$

Where $\mathcal{L}_{x\to y}$ can be a cross-entropy loss for a segmentation problem, or a MSE loss for a regression problem for example.

3.2 Intra-Domain Dependencies Regularization

When a pair of views is available as $(x, y) \sim p(x, y)$, or when only $x \sim p(x)$, or only $y \sim p(y)$ are available, it is possible to regularize the training of the model by adding two additional reconstruction losses. These losses based on the manifold assumption allow to learn intra-domain dependencies via the mean of reconstruction, with auto-encoder for example [3, 12, 13]. The model optimizes the reconstruct of the views from their latent code as:

$$\hat{x} = D_x(\tilde{x}) = D_x \circ E_x(x) \tag{3}$$

$$\hat{y} = D_y(\tilde{y}) = D_y \circ E_y(y) \tag{4}$$

The two associated reconstruction losses

$$\min_{\theta_{E_x},\theta_{D_x}} \mathbb{E}_{p(x)}\mathcal{L}_{x\to x}(D_x \circ E_x(x), x) = \mathcal{L}_{x\to x}(\hat{x}, x) \tag{5}$$

$$\min_{\theta_{E_y},\theta_{D_y}} \mathbb{E}_{p(y)}\mathcal{L}_{y\to y}(D_y \circ E_y(y), y) = \mathcal{L}_{y\to y}(\hat{y}, y) \tag{6}$$

can be used also when not all the data are available. When only x (resp. y) is available, only $\mathcal{L}_{x\to x}$ (resp. $\mathcal{L}_{y\to y}$) is optimized.

In machine learning, it is a common case to have plenty of unlabeled x only views, but to have only labels without the x available is more rare. For example, in a domain translation as style transfer, having y only sample is often easy to obtain as it needs collecting images of the desired style. For some other tasks like semantic segmentation, it is possible to obtain y only sample in road segmentation through the usage of simulator [27], in medical images segmentation the use of phantom able the generation of y only sample [8].

So far, by only taking the translation from X to Y, this framework corresponds to the SOP architecture [2]. LSM extends SOP in order to allow multi-domain translation by taking into account the transition from Y to X and further leveraging inter-domain dependencies.

3.3 Inter-Domain Dependencies Regularization

Unlike previous approaches using the shared latent space assumption (one can mention CorrNet [4] which our approach is similar to), we consider that this assumption should be relaxed and that $L_{\tilde{x}\to\tilde{y}}(\tilde{x})$ should be close to $\tilde{y}$. CorrNet makes the assumption of a fully shared latent space and applies correlation regularization between each latent code from different views. Without domain-specific information, the regularization might hinder the learning process by removing uncorrelated, domain-specific features which might be useful for the domain translation task. LSM prevents this issue by creating a latent space for each domain, and regularization is done between a latent code $\tilde{y}$ originating from a domain and a translated latent code into this domain $L_{\tilde{x}\to\tilde{y}}(\tilde{x})$. This does not penalize keeping domain-specific information in each latent code.

To enforce latent code $L_{\tilde{x}\to\tilde{y}}(\tilde{x})$ to be close to $\tilde{y}$, two additional losses are added, depending on the available supervision.

Supervised case In the case where the pair of view (x, y) is available, LSM minimizes

$$\min_{\theta_{E_x},\theta_{L_{\tilde{x}\to\tilde{y}}},\theta_{E_y}} \mathbb{E}_{p(\tilde{x},\tilde{y})}\mathcal{L}_{Dist_{\tilde{y}}}(\tilde{x}, \tilde{y}) = d(L_{\tilde{x}\to\tilde{y}}(\tilde{x}), \tilde{y}) \tag{7}$$

where d is a distance function, as for example L_1 distance. Minimizing $\mathcal{L}_{Dist_{\tilde{y}}}$ allows bringing closer the translated latent code $L_{\tilde{x}\to\tilde{y}}(\tilde{x})$ and the latent code $\tilde{y}$. In the supervised case, where all views are available, for example in a road semantic segmentation context considering the RGB image and its segmentation

map (x, y), and their latent representation $(\tilde{x}, \tilde{y})$. Supposing that $\tilde{y}$ allows reconstructing y through the decoder $D_y(\tilde{y}) = y$, then if $L_{\tilde{x}\to\tilde{y}}(\tilde{x}) = \tilde{y}$, obtaining $\tilde{y}$ using x would allow to translate x into y through the latent space mapping. In order for the translation to be accurate, one would expect $L_{\tilde{x}\to\tilde{y}}(\tilde{x})$ to be close to $E_y(y) = \tilde{y}$. Equation 7 enforce this constraint by minimizing the distance between latent codes.

Unsupervised case When the supervision is incomplete (either only x or only y is available), we substitute the distance loss $\mathcal{L}_{Dist_{\tilde{y}}}$ with a feature adversarial loss inspired by existing domain adaptation approaches as done in [34].

We would like the translated latent code $L_{\tilde{x}\to\tilde{y}}(\tilde{x})$ to be invariant from the latent code $\tilde{y}$. Classical machine learning models assume that all the input came from the same distribution, in this case the model is the decoder, and the inputs are latent codes. We consider latent codes to be invariant if a domain classifier is unable to distinguish its source. For that purpose, a domain classifier $DC_{\tilde{y}}$ is added on top of Y latent space in order to draw the two distributions closer. For the domain classifier, the loss is a binary cross-entropy (BCE) where the discriminator has to correctly classify the presented latent code in the right domain.

$$\min_{\theta_{DC_{\tilde{y}}}} \mathbb{E}_{p(\tilde{x}),p(\tilde{y})}\mathcal{L}_{DC_{\tilde{y}}}(\tilde{x}, \tilde{y}) = \text{BCE}(DC_{\tilde{y}} \circ L_{\tilde{x}\to\tilde{y}}(\tilde{x}), 0) + \text{BCE}(DC_{\tilde{y}}(\tilde{y}), 1) \quad (8)$$

The domain classifier attempt to discern from which domain the presented latent code is sampled.

Classical adversarial approaches make the generating part maximizing the error of $DC_{\tilde{y}}$, however as state in [35], the two distributions $L_{\tilde{x}\to\tilde{y}}(\tilde{x})$ and $\tilde{y}$ are changing, this could lead to an optimization problem. When the generating part is optimal, the domain classifier would only have to flip its sign in order to produce the correct prediction. This would result in oscillations in the optimization. Therefore, we encourage domains to produced features $L_{\tilde{x}\to\tilde{y}}(\tilde{x})$ and $\tilde{y}$ to be indistinguishable, using a confusion loss [34], where we compare the domain classifier decision against a uniform distribution:

$$\min_{\theta_{E_x},\theta_{E_y},\theta_{L_{\tilde{x}\to\tilde{y}}}} \mathbb{E}_{p(\tilde{x}),p(\tilde{y})}\mathcal{L}_{Conf_{\tilde{y}}}(\tilde{x}, \tilde{y}) = \text{BCE}\left(DC_{\tilde{y}} \circ L_{\tilde{x}\to\tilde{y}}(\tilde{x}), \frac{1}{2}\right) \quad (9)$$

$$+ \text{BCE}\left(DC_{\tilde{y}}(\tilde{y}), \frac{1}{2}\right) \quad (10)$$

3.4 Final Loss

Finally, the model is trained to optimize the weighted sum of all the previous losses. For a two-way X/Y domain translation ($x \to y$ and $y \to x$), the global

loss is

$$\min_{\substack{\theta_{E_x},\theta_{E_y},\theta_{L_{\tilde{x}\to\tilde{y}}},\\ \theta_{D_x},\theta_{D_y},\theta_{L_{\tilde{y}\to\tilde{x}}}}} \mathcal{L}_{Final} = \lambda_1\mathcal{L}_{x\to x} + \lambda_2\mathcal{L}_{y\to y} + \lambda_3^1\mathcal{L}_{x\to y} + \lambda_3^2\mathcal{L}_{y\to x}$$
$$+ \lambda_4(\mathcal{L}_{Dist_{\tilde{x}}} + \mathcal{L}_{Dist_{\tilde{y}}}) + \lambda_5(\mathcal{L}_{Conf_{\tilde{x}}} + \mathcal{L}_{Conf_{\tilde{y}}}) \quad (11)$$
$$\min_{\theta_{DC_{\tilde{x}}},\theta_{DC_{\tilde{y}}}} \mathcal{L}_{Adversarial} = \mathcal{L}_{DC_{\tilde{x}}} + \mathcal{L}_{DC_{\tilde{y}}} \quad (12)$$

It allows the model to receive supervision regardless of whether the samples are drawn from a marginal distribution $x \sim p(x)$ and $y \sim p(y)$, or from their joint distribution $x, y \sim p(x, y)$. $\mathcal{L}_{Dist_{\tilde{y}}}$ and $\mathcal{L}_{Conf_{\tilde{y}}}$ applied on a latent space on domain Y further push the model to create an embedding where the translated latent code $L_{\tilde{x}\to\tilde{y}}(\tilde{x})$ and the original latent code $\tilde{y}$ are more similar. In the general cases where all domains are similar (e.g. both domains are real images), $\lambda_3^1 = \lambda_3^2$.

This formulation differs from other models as CorrNet [4], which makes the assumption of the single shared latent space. Similarly to CorrNet, a representation is learned by means of reconstruction using autoencoders and a correlation regularization term in the latent space is applied. The main differences are that a) CorrNet modelizes a shared latent space as a combination of latent codes from different views. The unique shared latent code $\tilde{z}$ is constructed as $\tilde{z} = \sigma(W_x\tilde{x} + W_y\tilde{y} + b)$ where σ is an activation function, $\tilde{x} = E_x(x)$ and $\tilde{y} = E_y(y)$ are the last activation of the encoder for each domain, W and b are projections matrices and bias. This is the formulation when both views are available. When only one view is available (for example only x) this formulation becomes $\tilde{z} = \sigma(W_x\tilde{x} + b)$ as y is set to $\mathbf{0}$. b) CorrNet maximizes the Pearson correlation when pairs of data are available, this approach is classical in disentanglement learning, where the correlation (or mutual information) between domain-agnostic latent codes is maximized (section 2.3). In addition, in most disentanglement learning frameworks, a domain-specific latent code ensures that the specific information is kept, which is not the case in CorrNet. These optimization terms require all the views to be available at the same time.

We think that the formation introduced in a) might hinder the training, as by construction, given two views (x, y), the latent code $\tilde{z}$ has to be different if only x is available, or only y is available, or if the two views are available at the same time. This assumption may not hold in general, for example in a road context with two domains: (x) RGB images and (y) polarimetric images. We might expect the latent code $\tilde{x} = E_x(x)$ to be equal to $\tilde{y} = E_y(y)$, but also in the case where both views are available, equal to $\tilde{z} = E_{xy}(x, y)$, where $\tilde{z}$ is the denomination of the latent code when both views are available, and E_{xy} is a general encoding function. In CorrNet this is impossible by construction.

LSM tackles this by removing the shared latent space assumption without adding the separation of domain-specific and domain-agnostic latent code, and adding the mapping network. The reconstruction loss is then applied either on the origin latent code, or on the translated latent code in a similar fashion as in

CorrNet. Furthermore, LSM encourages latent code from different views to be similar using the distance and the latent discriminator.

b) We observe that CorrNet regularizes the latent space only when all the views are available. LSM leverages the information from the latent discriminator from the confusion adversarial loss (equation 8) in order to encourage the latent code from different views to be indistinguishable even when no supervision is available.

4 Experiments

In this section, we provide an evaluation of LSM along with three baselines on three datasets. To evaluate the benefits induced by LSM, we evaluate every approach using the same architecture that we made as simple as possible. We empirically show that the use of LSM is not detrimental when a lot of supervision is available. Furthermore, LSM allows for improving the training when additional unsupervised information is available and improves the training while reducing the variance of the models. Qualitative results are available in Appendix D.

4.1 Datasets

Synthetic Image Translation (SIT) Dataset The synthetic image translation dataset consists of image pairs made by superposing a background texture and a ring from another texture [2] for the input domain and the associated circle segmentation for the output domain. Four parameters control the texture generation: circle x position, circle y position, circle radii, and circle thickness. In order to complexify the task, for the segmentation mask, we swapped the x position with the circle thickness and the y position with the circle radii. This changes the task from a segmentation task to a more challenging image-to-image translation task. We use the standard mean Intersection over Union (mIoU) as evaluation metric for the output prediction. Figure 4 displays two pairs of an input image and ground truth. We refer to Appendix A.1 for dataset description details.

Facial Landmark Detection (300-W) Dataset We study the real case of facial landmark detection, on the 300 Faces In-the-Wild Challenge (300-W) [30, 31]. This 300 Faces In-the-Wild Challenge dataset is constituted of 300 indoor images and 300 outdoor images with associated 68 facial landmarks for each face. It provides a context with multiple modalities (image faces, and vector of positions) and with a highly structured output domain that presents strong intra-dependencies. We use the standard Normalized Root Mean Square Error (NRMSE) as metric for facial landmark detection. We refer to Appendix A.2 and [30,31] for dataset description details, examples of data are given in Figure 5.

[2] https://www.ux.uis.no/∼tranden/brodatz.html

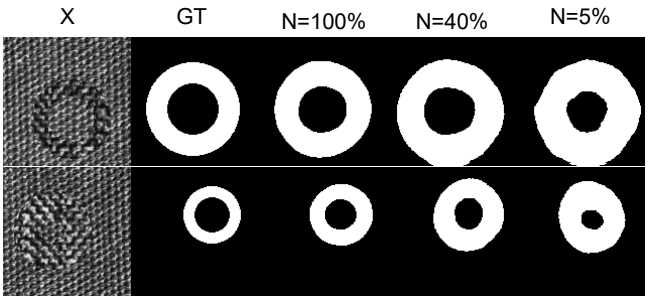


Fig. 4: SIT samples with their associated prediction from LSM for different amounts of available (x, y) views. More predictions in Appendix D

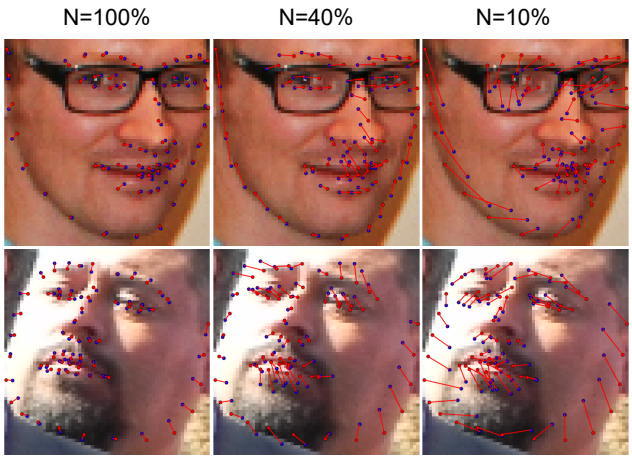


Fig. 5: 300-W with their associated facial landmarks (red dot) and predicted landmark from LSM (blue dot) for different amounts of available (x, y) views. More predictions in Appendix D

Sarcopenia Semantic Segmentation Dataset (SSS) We study segmentation of the third lumbar vertebra (L3) level from Computed Tomography (CT) scans. This task is needed for predicting the sarcopenia level, which is a measure of muscle atrophy, later used in cancer diagnostic [19, 26]. We use the mean Intersection over Union (mIoU) as metric for the segmentation. We refer to Appendix A.3 for dataset description details, examples of data are given in Figure 6.

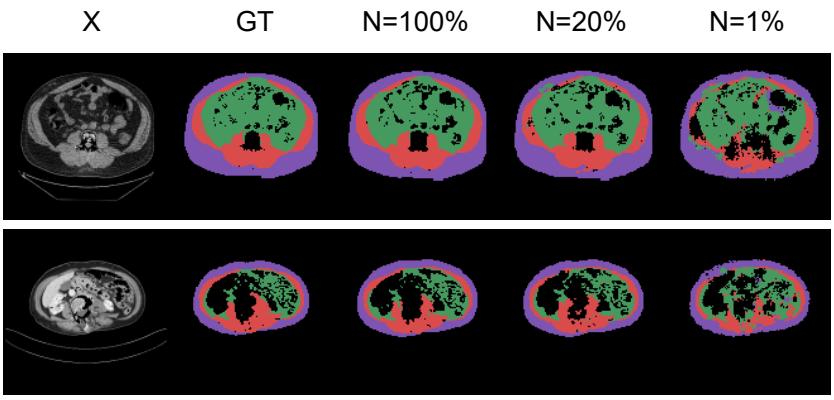


Fig. 6: SSS samples with their associated prediction from LSM for different amounts of available (x, y) views. More predictions in Appendix D

4.2 Baselines

We compare our model with 5 baselines: a basic baseline, IODA [21], SOP [2], CycleGAN [38] an unsupervised domain translation approach, and Pix2Pix [17] a supervised domain translation approach. Every baseline used has the same model architecture. For the architecture, we used this[3] implementation from GitHub designed for CycleGAN and Pix2Pix. When necessary, the models have been adapted for the use case.

The **basic baseline**, is restricted to $D_y \circ L_{\tilde{x}\to\tilde{y}} \circ E_x$ as in Section 3.1, this corresponds to using only the input to output translation with the loss described in equation 2. **IODA** [21] model has a similar structure as our model and also exploits information from input data and output data, but the learning of additional information is performed offline during a pre-training phase. Namely IODA first pre-trains $D_x \circ E_x$ and $D_y \circ E_y$. Then in a final phase, the model $D_y \circ L_{\tilde{x}\to\tilde{y}} \circ E_x$ is initialized using the pre-trained weights and training using equation 2. **SOP** [2] has a similar architecture, and uses information from input and output as a mean to regularize the training of equation 2. The difference between the SOP model and our model lies in the equation 7 and equation 8 that

[3] https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

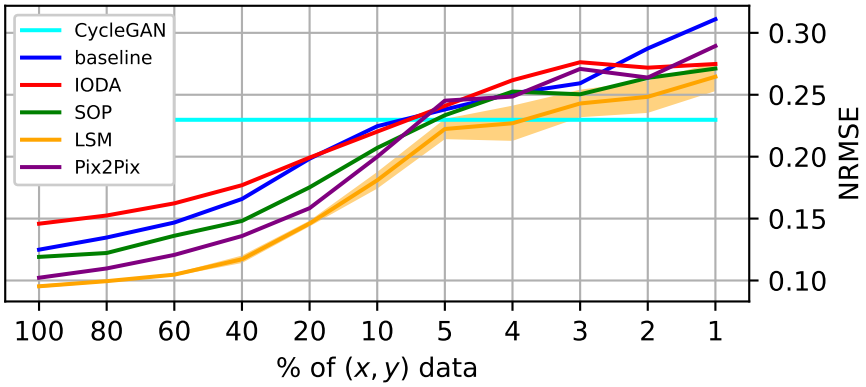
are not present in SOP. Namely, a distance cost brings closer representations from different origins, and an adversarial cost pushes closer different embedding to follow the same distribution. **CycleGAN** [38] learns a two-ways domain translation in an unsupervised way. Unlike IODA, SOP, and LSM, it does not explicitly modelizes a latent space. The translation $x \to y$ is learned by a generator $D_y \circ L_{\tilde{x}\to\tilde{y}} \circ E_x$ and is enforced realistic using a domain discriminator and a meaningful semantic is enforced using a cycle-consistency loss between x and $D_x \circ L_{\tilde{y}\to\tilde{x}} \circ E_y \circ D_y \circ L_{\tilde{x}\to\tilde{y}} \circ E_x$). **Pix2Pix** [17] learns a one-way domain translation in a supervised way. Not explicitly modelizing a latent space, Pix2Pix learns a generator $D_y \circ L_{\tilde{x}\to\tilde{y}} \circ E_x$. The supervision came from a discriminator that takes as input either (x, y) as the real examples or $(x, D_y \circ L_{\tilde{x}\to\tilde{y}} \circ E_x(y))$ as the fakes examples, with an additional supervised loss between y and $D_y \circ L_{\tilde{x}\to\tilde{y}} \circ E_x(y)$.

4.3 Adapting to Lower Supervision

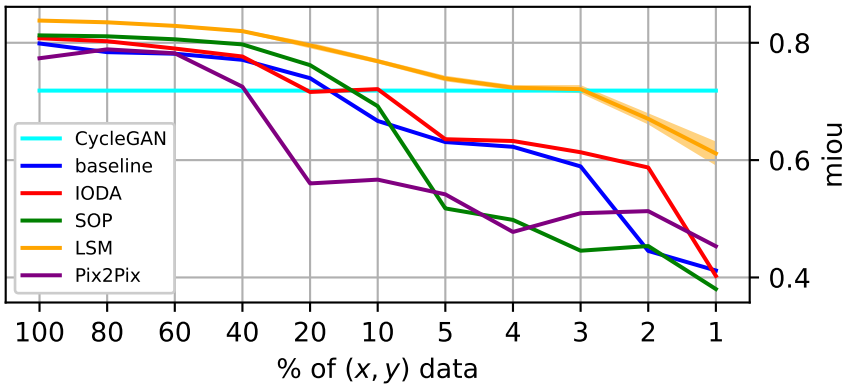
We first compare each model while adapting the amount of available supervision, controlled by taking $N\%$ of the dataset as supervised pair (x, y), and splitting evenly the remaining $(100 - N)\%$ as only x samples and only y samples (unsupervised pairs). We lower N in order to see an evolution in the metrics. To ensure that no x and y in the unsupervised set could come from the same pair, we divide the dataset in two before the training, keeping one part as a bank of unsupervised pairs. This has for effect to divide the effective dataset length by two. As CycleGAN operates in an unsupervised setting, its performance does not change for each N value. LSM is compared for different values of N for the facial landmark in Figure 7a, the sarcopenia dataset in Figure 7b and the synthetic image translation in Figure 8.

High Amount of Supervision For the highest N values ($N = 100$) and the 300-W and SSS dataset, there are no significant differences between models, as there is plenty of (x, y) views, the different baselines are able to learn the tasks. For SIT we found that LSM greatly improves on the metric compared to the baselines, this is explained by the fact that, unlike in the two previous tasks, there is not direct pixel-to-pixel (or spatial) correspondence between the two domains (due to the swapping of variables). We found that CycleGAN is able to perform on the 300-W and SSS tasks, where there is a pixel-to-pixel correspondence, but fails the SIT image translation, as the correlation between domains is not pixel-to-pixel, this unsupervised approach is not able to understand the swapping of variables and not suited for this task.

Medium values of N($N \in \{80, 60, 40\}$ for facial landmark, $N \in \{80, 60, 40\}$ for sarcopenia datasets, $N \in \{80\}$ for synthetic dataset) only slightly affect the training. For the two first, the metric only starts to really decrease for LSM and the baseline when $N < 40$, this is explained by the fact that the model does not need every sample to learn the task. Those values of N are Pareto optimal for this combination of dataset and model leading to either no or a slight decrease in the metrics. For the synthetic dataset SIT, the metric start to decrease earlier as the task might be harder for the models and the amount of data is lower.



(a)



(b)

Fig. 7: Impact of the number of available pairs from (a) the 300-W dataset on the NRMSE metric (lower is better) and (b) the SSS dataset on the mIoU metric (higher is better). Areas around LSM line represent the standard deviation (5 trained models for each baseline)

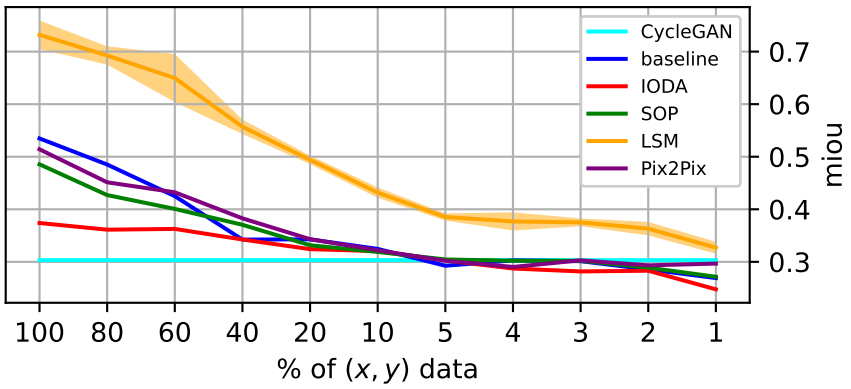


Fig. 8: Impact of the number of available pairs from the synthetic image translation dataset on the mIoU metric (higher is better). Areas around LSM line represent the standard deviation (5 trained models for each baseline)

Low Amount of Supervision When N became low ($N < 60$ for every dataset) and the amount of supervision became limited, every model's performance start to degrade as the amount of supervision is no longer sufficient to completely learn the task and generalize over the test set.

Table 1: NRMSE evaluation on the Facial Landmark Dataset according to the portion of available pairs (x, y) from the dataset

Model	100 %	60 %	10 %	5 %	4 %	3 %	2 %	1 %
Baseline	1.25e-1	1.47e-1	2.25e-1	2.38e-1	2.51e-1	2.59e-1	2.87e-1	3.11e-1
IODA	1.46e-1	1.62e-1	2.20e-1	2.41e-1	2.62e-1	2.76e-1	2.72e-1	2.75e-1
SOP	1.19e-1	1.36e-1	2.07e-1	2.33e-1	2.53e-1	2.50e-1	2.64e-1	2.71e-1
LSM	**9.53e-2**	**1.05e-1**	**1.81e-1**	**2.22e-1**	**2.27e-1**	*2.43e-1*	*2.48e-1*	*2.65e-1*
Pix2Pix	*1.02e-1*	*1.21e-1*	*2.00e-1*	2.45e-1	2.49e-1	2.71e-1	2.64e-1	2.89e-1
CycleGAN	2.30e-1	2.30e-1	2.30e-1	*2.30e-1*	*2.30e-1*	**2.30e-1**	**2.30e-1**	**2.30e-1**

For the 300-W and SIT datasets (Figure 7a) and the lowest amount of supervision ($N = 1$), the model performances decline and reach the same NRMSE as the supervised and unsupervised baselines. While for the SSS dataset, LSM is able to better adapt to a low amount of supervision than the other supervised baselines. In the low-supervision setting, unsupervised models such as CycleGAN are able to perform well, for $N \leq 4$ for 300-W, and $N \leq 2$ for SSS. For the synthetic dataset, the task is impossible to solve without supervision due to the swapping of variable, and LSM stay above all the baseline for low values of N.

For the sarcopenia dataset Figure 7b, we can note that LSM performances decrease slower than the supervised baselines. We conjuncture that LSM is able to leverage the side information from only x or only y views without increasing the discrepancy between both domains. Where SOP learning schema does not take into account this discrepancy and falls behind the baseline. We suppose that IODA is able to perform well in this kind of setting by providing a good parameter initialization without constraining the later learning of the model.

Table 2: mIoU evaluation on the synthetic image translation dataset according to the portion of available pairs (x, y) from the dataset

Model	100 %	60 %	10 %	5 %	4 %	3 %	2 %	1 %
Baseline	*5.35e-1*	4.25e-1	*3.25e-1*	2.93e-1	3.03e-1	3.01e-1	2.86e-1	2.69e-1
IODA	3.74e-1	3.63e-1	3.20e-1	3.03e-1	2.87e-1	2.82e-1	2.83e-1	2.48e-1
SOP	4.86e-1	4.01e-1	3.19e-1	*3.05e-1*	3.02e-1	3.02e-1	2.89e-1	2.72e-1
LSM	**7.32e-1**	**6.50e-1**	**4.32e-1**	**3.85e-1**	**3.77e-1**	**3.75e-1**	**3.63e-1**	**3.27e-1**
Pix2Pix	5.14e-1	*4.32e-1*	3.22e-1	3.02e-1	2.90e-1	3.03e-1	2.93e-1	2.97e-1
CycleGAN	3.03e-1	3.03e-1	3.03e-1	3.03e-1	*3.03e-1*	*3.03e-1*	*3.03e-1*	*3.03e-1*

Table 3: mIoU evaluation on the sarcopenia dataset according to the portion of available pairs (x, y) from the dataset

Model	100 %	60 %	10 %	5 %	4 %	3 %	2 %	1 %
Baseline	7.99e-1	7.81e-1	6.67e-1	6.31e-1	6.23e-1	5.89e-1	4.45e-1	4.12e-1
IODA	8.08e-1	7.90e-1	*7.21e-1*	6.36e-1	6.33e-1	6.13e-1	5.87e-1	4.03e-1
SOP	*8.13e-1*	*8.06e-1*	6.92e-1	5.18e-1	4.98e-1	4.46e-1	4.54e-1	3.80e-1
LSM	**8.38e-1**	**8.29e-1**	**7.69e-1**	**7.39e-1**	**7.23e-1**	**7.21e-1**	*6.70e-1*	*6.11e-1*
Pix2Pix	7.74e-1	7.82e-1	5.67e-1	5.42e-1	4.78e-1	5.09e-1	5.13e-1	4.53e-1
CycleGAN	7.18e-1	7.18e-1	7.18e-1	*7.18e-1*	*7.18e-1*	*7.18e-1*	**7.18e-1**	**7.18e-1**

5 Conclusions

We presented LSM, a novel training framework allowing to learn domain translation with multiple supervision levels, either fully supervised, or with views from only some domains. LSM learns for each domain a latent space and a mapping function between each of these latent spaces. Unlike previous approaches, LSM does not make the shared latent assumption directly, but rather relaxes it by supposing the existence of a mapping function between latent spaces. And

further constraints each mapped latent code to be consistent with the original latent code from one domain. This makes LSM suitable for multi-modal domain translation tasks, especially when incomplete supervision is available.

The experiments confirmed that LSM is not penalizing when a lot of supervision is available and is able to leverage additional information in order to improve the training of models in low data settings.

While LSM allows learning the translation between multiple domains in a weakly-supervised setting, it does not leverage the availability of multiple related views to enhance the translation during the inference (LSM leverages the availability of multiple related views during training only). We plan to address this issue in future works which will combine information from multiple related views as input to improve the translation. Another issue that arises from LSM is the increasing model size as the number of domains grows, this makes such an approach heavy for a high number of domains.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan (2017)
2. Belharbi, S., Chatelain, C., Hérault, R., Adam, S.: Input/output deep architecture for structured output problems. CoRR **abs/1504.07550** (2015)
3. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., Montreal, U.: Greedy layer-wise training of deep networks. Advances in Neural Information Processing Systems **19** (01 2007)
4. Chandar, S., Khapra, M.M., Larochelle, H., Ravindran, B.: Correlational neural networks (2015)
5. Chapelle, O., Scholkopf, B., Zien, Eds., A.: Semi-supervised learning (chapelle, o. et al., eds.; 2006) [book reviews]. IEEE Transactions on Neural Networks **20**(3), 542–542 (2009)
6. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation (2018)
7. Cristinacce, D., Cootes, T.: Feature detection and tracking with constrained local models. In: BMVC (2006)
8. Fedrigo, R., Segars, W.P., Martineau, P., Gowdy, C., Bloise, I., Uribe, C.F., Rahmim, A.: Development of the lymphatic system in the 4d xcat phantom (2021)
9. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation (2015)
10. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks (2014)
11. Grover, A., Chute, C., Shu, R., Cao, Z., Ermon, S.: Alignflow: Cycle consistent learning from multiple domains via normalizing flows (2019)
12. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
13. Hinton, G., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural computation **18**, 1527–54 (08 2006)
14. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation (2017)
15. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation (2016)

16. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation (2018)
17. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks (2018)
18. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2014)
19. Lanic, H., Kraut-Tauzia, J., Modzelewski, R., Clatot, F., Mareschal, S., Picquenot, J.M., Stamatoullas, A., Leprêtre, S., Tilly, H., Jardin, F.: Sarcopenia is an independent prognostic factor in elderly patients with diffuse large B-cell lymphoma treated with immunochemotherapy. Leukemia lymphoma **55**(4), 817–23 (Mar 2014)
20. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M.K., Yang, M.H.: Diverse image-to-image translation via disentangled representations (2018)
21. Lerouge, J., Herault, R., Chatelain, C., Jardin, F., Modzelewski, R.: Ioda: An input/output deep architecture for image labeling. Pattern Recognition **48**(9), 2847–2858 (2015)
22. Li, H., Wan, R., Wang, S., Kot, A.: Unsupervised domain adaptation in the wild via disentangling representation learning. International Journal of Computer Vision **129** (02 2021)
23. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
24. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks (2018)
25. Liu, Y., Nadai, M.D., Yao, J., Sebe, N., Lepri, B., Alameda-Pineda, X.: Gmm-unit: Unsupervised multi-domain and multi-modal image-to-image translation via attribute gaussian mixture modeling (2020)
26. Martin, L., Birdsell, L., MacDonald, N., Reiman, T., Clandinin, M.T., McCargar, L.J., Murphy, R., Ghosh, S., Sawyer, M.B., Baracos, V.E.: Cancer cachexia in the age of obesity: Skeletal muscle depletion is a powerful prognostic factor, independent of body mass index. Journal of Clinical Oncology **31**(12), 1539–1547 (2013)
27. Martinez, M., Sitawarin, C., Finch, K., Meincke, L., Yablonski, A., Kornhauser, A.L.: Beyond grand theft auto V for training, testing and enhancing deep learning in self driving cars. CoRR **abs/1712.01397** (2017)
28. Pei, Z., Cao, Z., Long, M., Wang, J.: Multi-adversarial domain adaptation (2018)
29. Reiß, S., Seibold, C., Freytag, A., Rodner, E., Stiefelhagen, R.: Every annotation counts: Multi-label deep supervision for medical image segmentation. CoRR **abs/2104.13243** (2021)
30. Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: database and results. Image and Vision Computing **47**, 3–18 (2016), 300-W, the First Automatic Facial Landmark Detection in-the-Wild Challenge
31. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: 2013 IEEE International Conference on Computer Vision Workshops. pp. 397–403 (2013)
32. Sun, H., Mehta, R., Zhou, H.H., Huang, Z., Johnson, S.C., Prabhakaran, V., Singh, V.: Dual-glow: Conditional flow-based generative model for modality transfer (2019)
33. Thanh-Tung, H., Tran, T.: On catastrophic forgetting and mode collapse in generative adversarial networks (2020)
34. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks (2015)

35. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation (2017)
36. Vu, H.T., Huang, C.C.: Domain adaptation meets disentangled representation learning and style transfer (2018)
37. Wu, Z., Han, X., Lin, Y.L., Uzunbas, M.G., Goldstein, T., Lim, S.N., Davis, L.S.: Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation (2018)
38. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks (2020)

A Dataset Details

A.1 Synthetic Image Translation (SIT) Dataset

The synthetic image translation dataset consists of image pairs made by superposing a background texture and a ring from another texture [4] for the input domain and the associated circle segmentation for the output domain. Four parameters control the texture generation: circle x position, circle y position, circle radii, and circle thickness. In order to complexify the task, for the segmentation mask, we swapped the x position with the circle thickness and the y position with the circle radii. This changes the task from a segmentation task to a more challenging image-to-image translation task. We use the standard mean Intersection over Union (mIoU) as evaluation metric for the output prediction and generate a total of 500 images. Figure 4 displays two pairs of an input image and ground truth. The images are 128×128 pixels (we randomly take a square in the texture 17 and 77 of size 128×128). Values are normalized in $[0, 1]$, 500 samples are generated. We train on 70% of the data, keep 10% for the validation and 20% for the testing. We use the cross-entropy as the loss function for the segmentation-related tasks, mean-square-error for input reconstruction-related tasks. As segmentation metric, we use the mean intersection-over-union, computed on the three classes and not on the background.

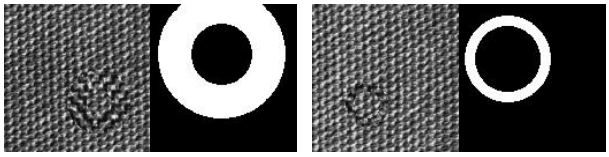


Fig. 9: Two couples of input image textures and their associated output. Here we swap generating parameter between the image and its segmentation creating an image-to-image translation task

A.2 Facial Landmark Detection (300-W) Dataset

We study the real case example of facial landmark detection, on the 300 Faces In-the-Wild Challenge (300-W) [30, 31]. This 300 Faces In-the-Wild Challenge dataset is constituted of 300 indoor images and 300 outdoors images with associated 68 facial landmarks for each face. It provides a context with multiple modalities (image faces, and vector of positions) and with a highly structured output domain that presents strong intra-dependencies.

We rescale each image into 64×64 pixels and normalize the pixels values in $[0, 1]$ and the landmarks in $[0, 1]$. We train on 70% of the data, keep 10% for

[4] https://www.ux.uis.no/∼tranden/brodatz.html

the validation and 20% for the testing. We use the mean-square error for input reconstruction-related tasks and prediction-related tasks. We use Normalized Root Mean Squared Error (NRMSE in equation 13) [7] as the metric for the prediction task.

$$NRMSE(\hat{y}, y) = \frac{\sum_{i=1}^{N} \|\hat{y}_i - y_i\|_2}{N \times D} \tag{13}$$

With $N = 68$ the number of landmark points $\hat{y} \in \mathbb{R}^{N\times 2}$ is the landmark prediction and $y \in \mathbb{R}^{N\times 2}$ is the landmark ground-truth, and D is the inter-ocular distance computed from y.

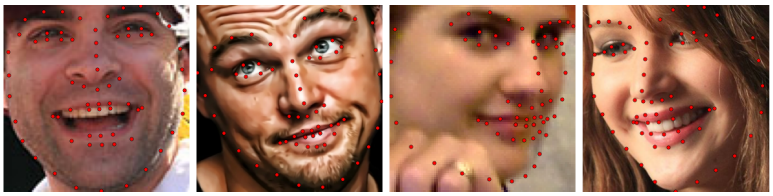


Fig. 10: Faces with their associated facial landmarks from the 300-W dataset

A.3 Sarcopenia Semantic Segmentation Dataset (SSS)

Segmentation of the third lumbar vertebra (L3) level is useful in order to predict, from CT scan (computed tomography scan), the sarcopenia level - characterizing a level of muscle atrophy - which is later used in cancer diagnostic [19, 26]. The segmentation task is a tedious task for the medical practitioner as the hand-made segmentation can take 4 minutes for an expert [21]. Two challenges arise from the data, inter-patient variability, as there are different factors that will impact the CT scan (age, sex, disease ...); and inter-image variability, a lot of parameters exterior the patient can impact the CT scan quality (quality of the device, the radiation dosage ...). This is illustrated in Figure 6 with two samples of CT scans and their ground-truth segmentation that are really different from one another. Examples of data are given in Figure 6, there are 4 classes: background (black), subcutaneous adipose tissue (purple), visceral adipose tissue (green), and skeletal muscle (red). The skeletal muscle segmentation is used to compute the sarcopenia level. The dataset contains a total of 527 pairs of CT images and their masks, we train on 70% of the data, keeping 10% for the validation and 20% for the testing. For this task, the image's background is cropped, they are resized to 128×128 pixels and normalized between 0 and 1. We use the cross-entropy as the loss function for the segmentation-related tasks, mean-square-error for input reconstruction-related tasks. As segmentation metric, we use the mean intersection-over-union, computed on the three classes and not on the background.

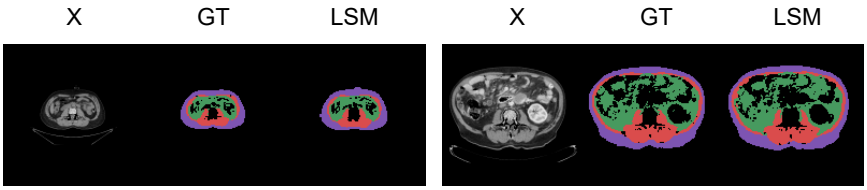


Fig. 11: Two tuples of CT scans, their ground-truth segmentation mask from our dataset and the LSM translation

B LSM Training complexity

We study the possible training overhead that LSM might induce in terms of the number of epochs. For each model, a training budget of 1000 epochs was available and for IODA, we also allow 1000 epochs for each pre-training stage. We compare how the different approaches behave within this computational budget and for each dataset. We display results in Figure 12.

We did not find that LSM uses an excessive number of epochs before early stopping (with a patience of 100 epochs), especially when compared to pre-training methods such as IODA, with an additional training stage, which can take a lot of training.

For methods using Generative Adversarial Networks such as CycleGAN and Pix2Pix, we consume all the training budget but display the best epoch per run. For unsupervised method as CycleGAN, it is not possible to perform early stopping as there is no ground truth available to compute a metric on an unpaired dataset in practice.

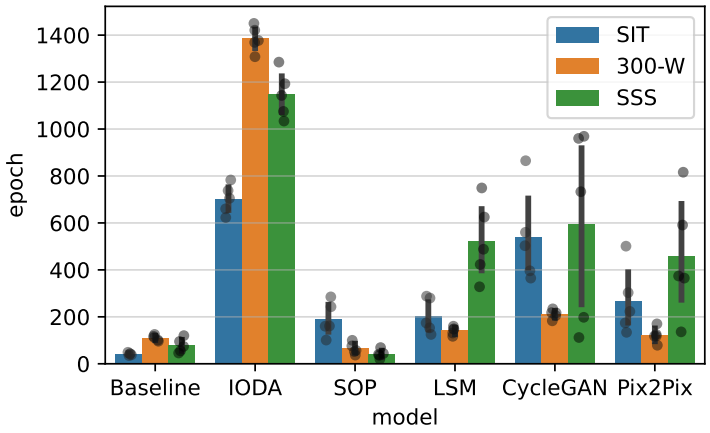


Fig. 12: Indices of the best epoch per model and per dataset using the fully labeled dataset ($N = 100$).

C Distance and Adversarial Loss Ablation

We study how the different losses contribute to the model performances while varying the amount of available supervision. We name each ablation according to the format $\Lambda_1\ \Lambda_2\ \Lambda_3\ \Lambda_4$ where $\Lambda_i = 1$ if $\lambda_i \neq 0$. We found that, among all configurations, the one using every loss, have the lower NRMSE and the higher mIoU.

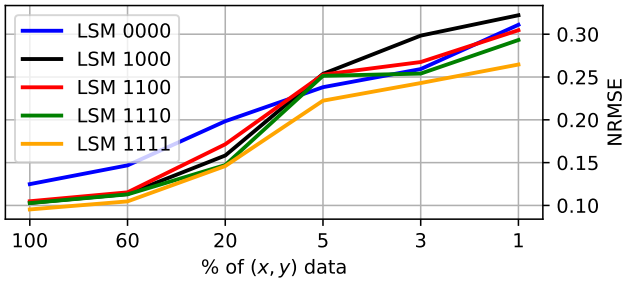


Fig. 13: Impact of the losses on LSM on 300-W dataset on the NRMSE metric

Table 4: Impact of losses ablation on LSM, on 300-W dataset on the NRMSE metric

Model	100 %	60 %	20 %	5 %	3 %	1 %
LSM 0000	1.25e-1	1.47e-1	1.98e-1	*2.38e-1*	2.59e-1	3.11e-1
LSM 1000	1.03e-1	1.13e-1	1.58e-1	2.54e-1	2.98e-1	3.22e-1
LSM 1100	1.05e-1	1.15e-1	1.71e-1	2.53e-1	2.67e-1	3.05e-1
LSM 1110	*1.03e-1*	*1.13e-1*	*1.47e-1*	2.51e-1	*2.54e-1*	*2.93e-1*
LSM 1111	**9.53e-2**	**1.05e-1**	**1.46e-1**	**2.22e-1**	**2.43e-1**	**2.65e-1**

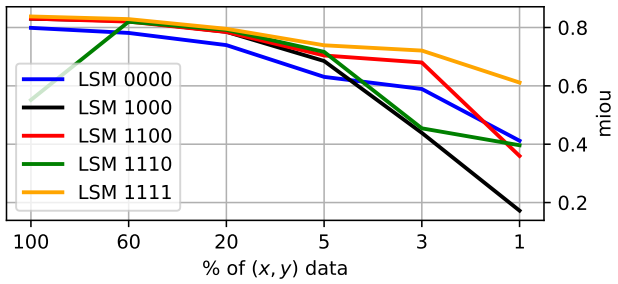


Fig. 14: Impact of the losses on LSM on SSS dataset on the mIoU metric

Table 5: Impact of losses ablation on LSM, on SSS dataset on the mIoU metric

Model	100 %	60 %	20 %	5 %	3 %	1 %
LSM 0000	7.99e-1	7.81e-1	7.40e-1	6.31e-1	5.89e-1	*4.12e-1*
LSM 1000	8.30e-1	*8.23e-1*	7.85e-1	6.85e-1	4.39e-1	1.72e-1
LSM 1100	*8.31e-1*	8.20e-1	7.85e-1	7.04e-1	*6.80e-1*	3.60e-1
LSM 1110	5.52e-1	8.20e-1	*7.92e-1*	*7.16e-1*	4.54e-1	3.96e-1
LSM 1111	**8.38e-1**	**8.29e-1**	**7.96e-1**	**7.39e-1**	**7.21e-1**	**6.11e-1**

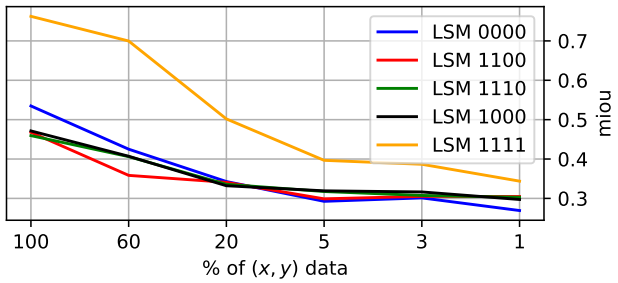


Fig. 15: Impact of the losses on LSM on SIT dataset on the mIoU metric

D Qualitative Results

We provide LSM prediction for each dataset and different levels of supervision to access the evolution of prediction quality according to the available supervision.

Table 6: Impact of losses ablation on LSM, on SIT dataset on the mIoU metric

Model	100 %	60 %	20 %	5 %	3 %	1 %
LSM 0000	*5.35e-1*	*4.25e-1*	*3.43e-1*	2.93e-1	3.01e-1	2.69e-1
LSM 1000	4.08e-1	3.76e-1	3.26e-1	3.16e-1	*3.04e-1*	*2.91e-1*
LSM 1100	3.99e-1	3.56e-1	3.26e-1	2.96e-1	3.00e-1	2.88e-1
LSM 1110	4.03e-1	3.73e-1	3.23e-1	*3.16e-1*	3.00e-1	2.90e-1
LSM 1111	**7.32e-1**	**6.50e-1**	**4.94e-1**	**3.85e-1**	**3.75e-1**	**3.27e-1**

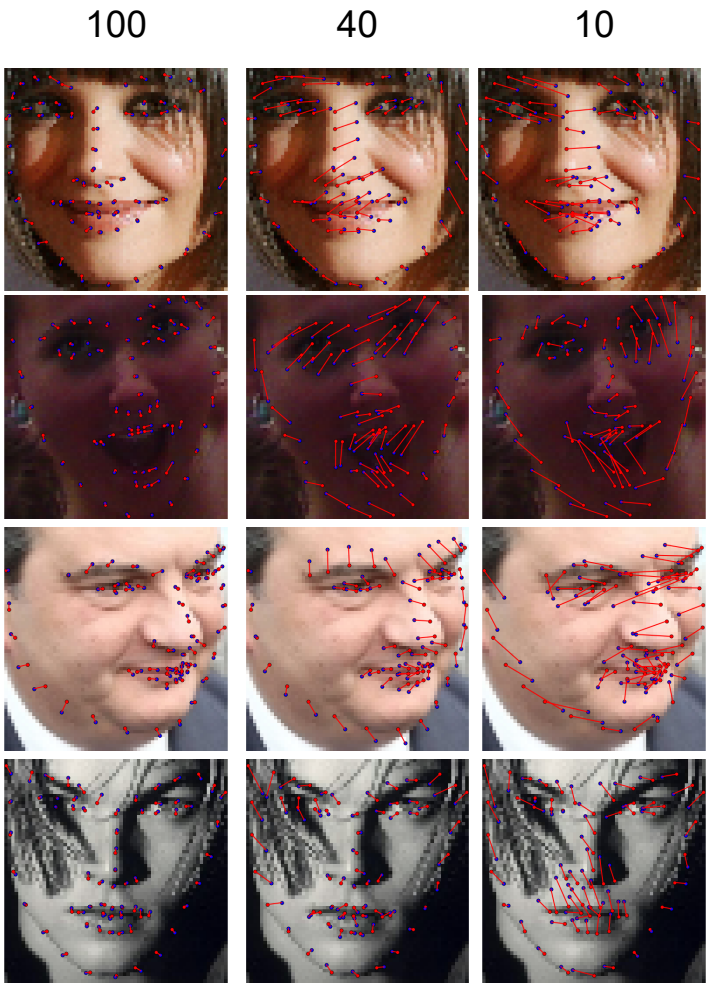


Fig. 16: Predicted sample from LSM on 300-W test set, with different proportion of (x, y) available (100%, 40%, 10%). Ground-truth marker in blue, prediction marker in red

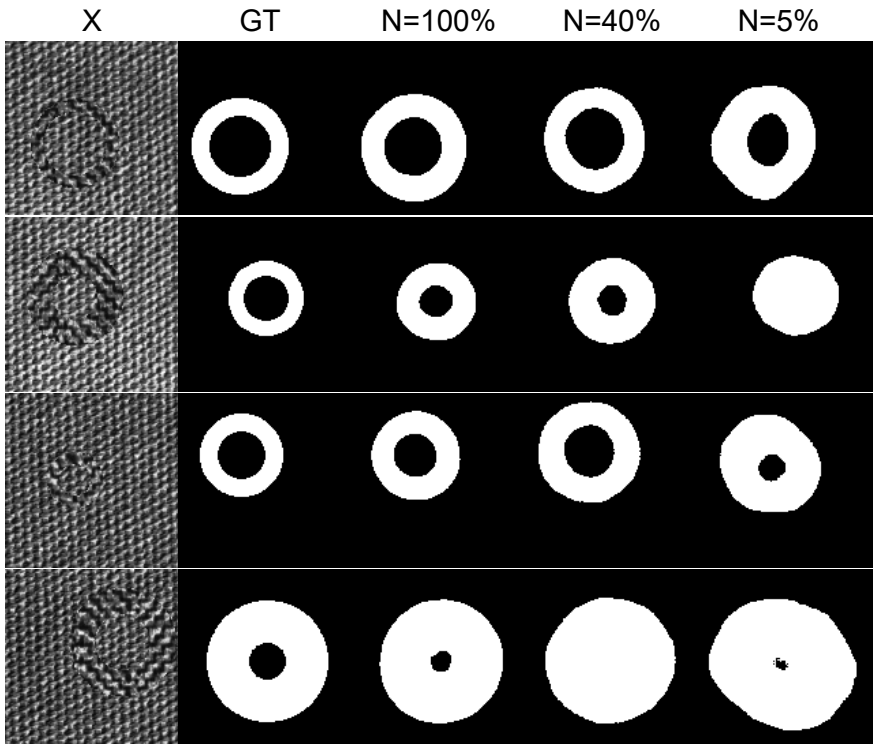


Fig. 17: Predicted sample from LSM on SIT test set, with different proportion of (x, y) available (100%, 40%, 5%)

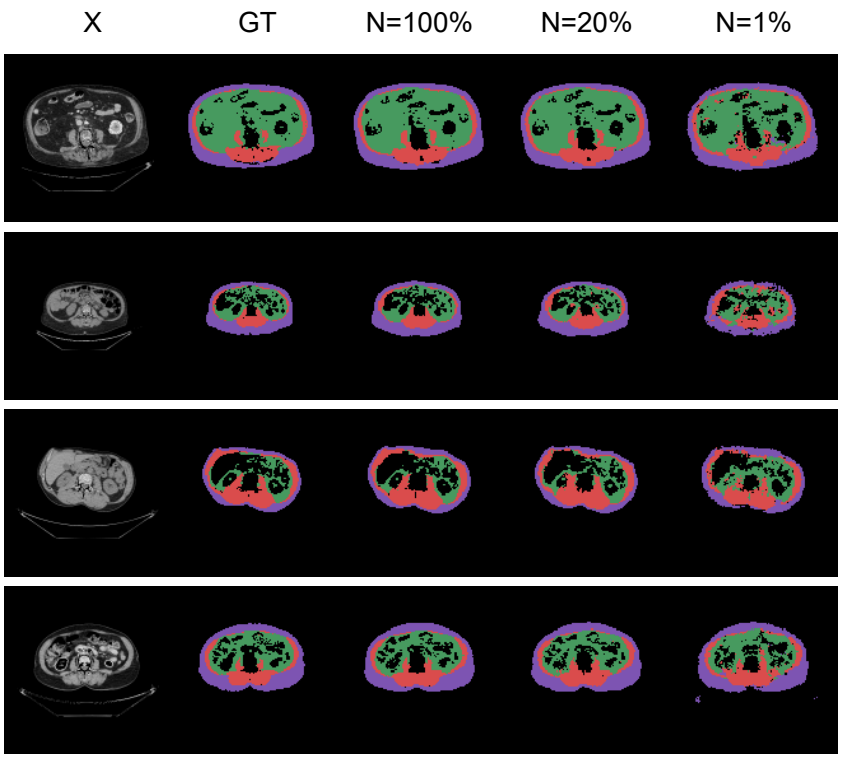


Fig. 18: Predicted sample from LSM on SSS test set, with different proportion of (x, y) available (100%, 20%, 1%)

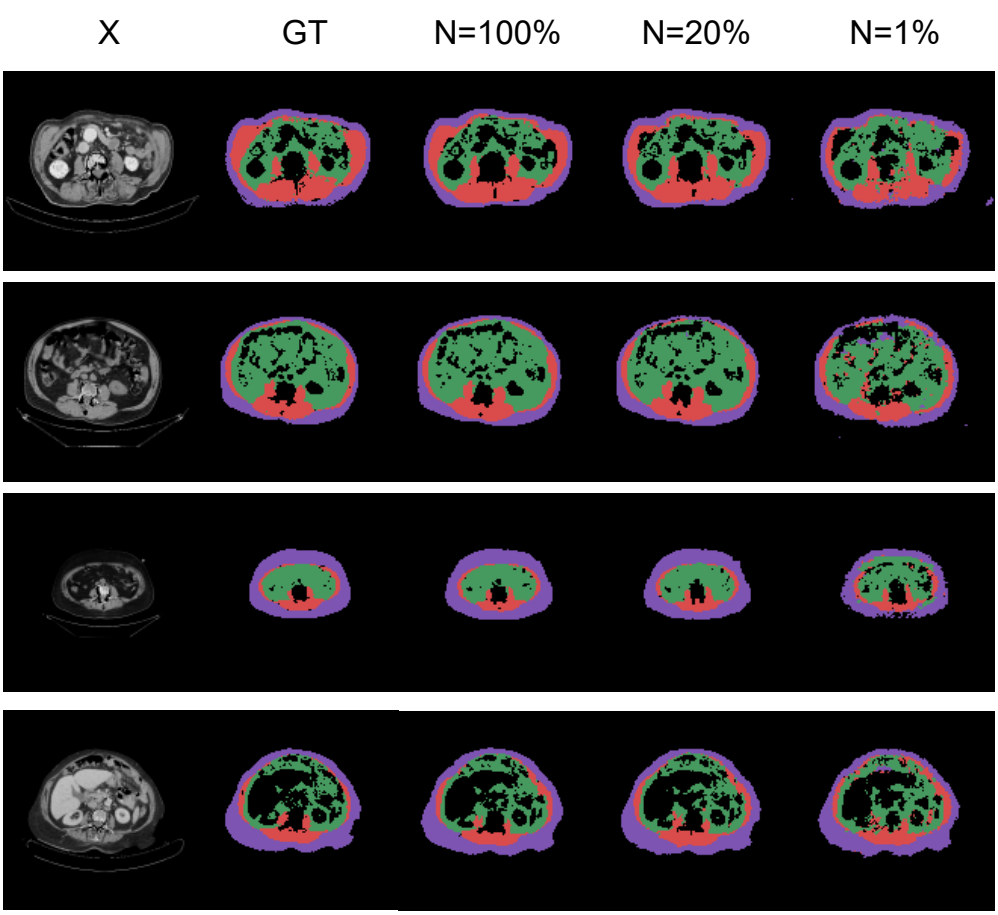


Fig. 19: Predicted sample from LSM on SSS test set, with different proportion of (x, y) available (100%, 20%, 1%)

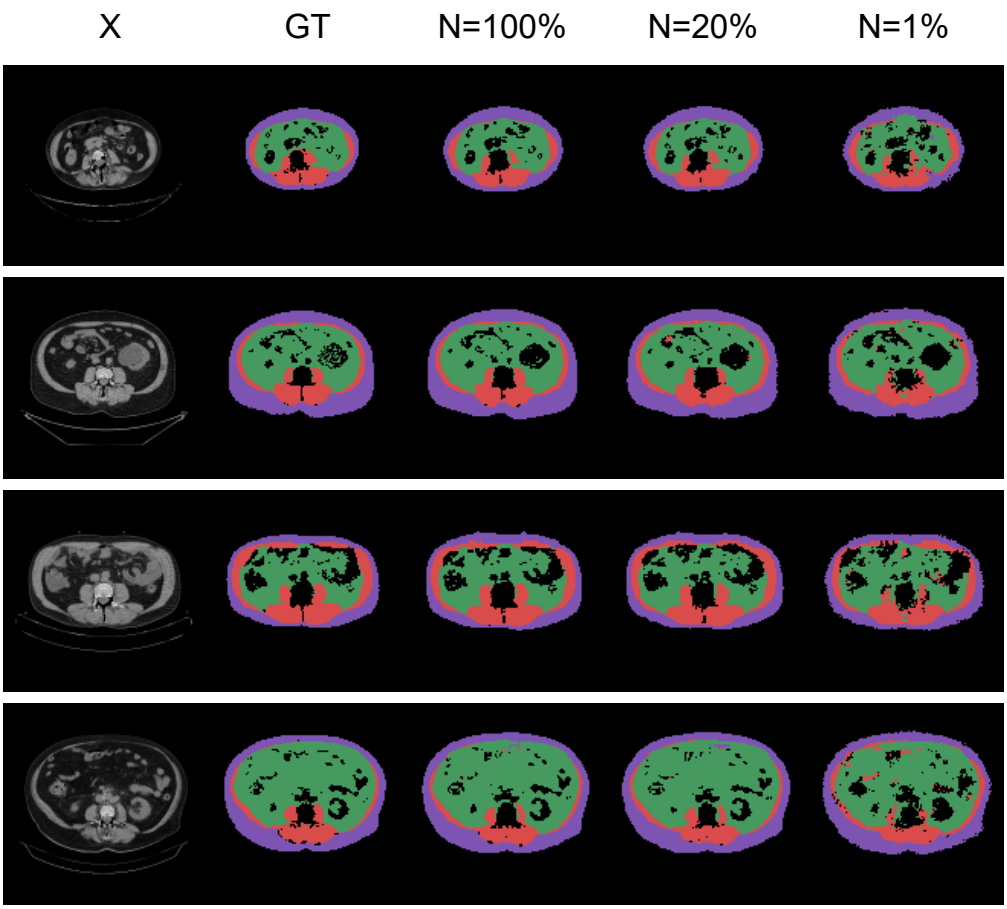


Fig. 20: Predicted sample from LSM on SSS test set, with different proportion of (x, y) available (100%, 20%, 1%)

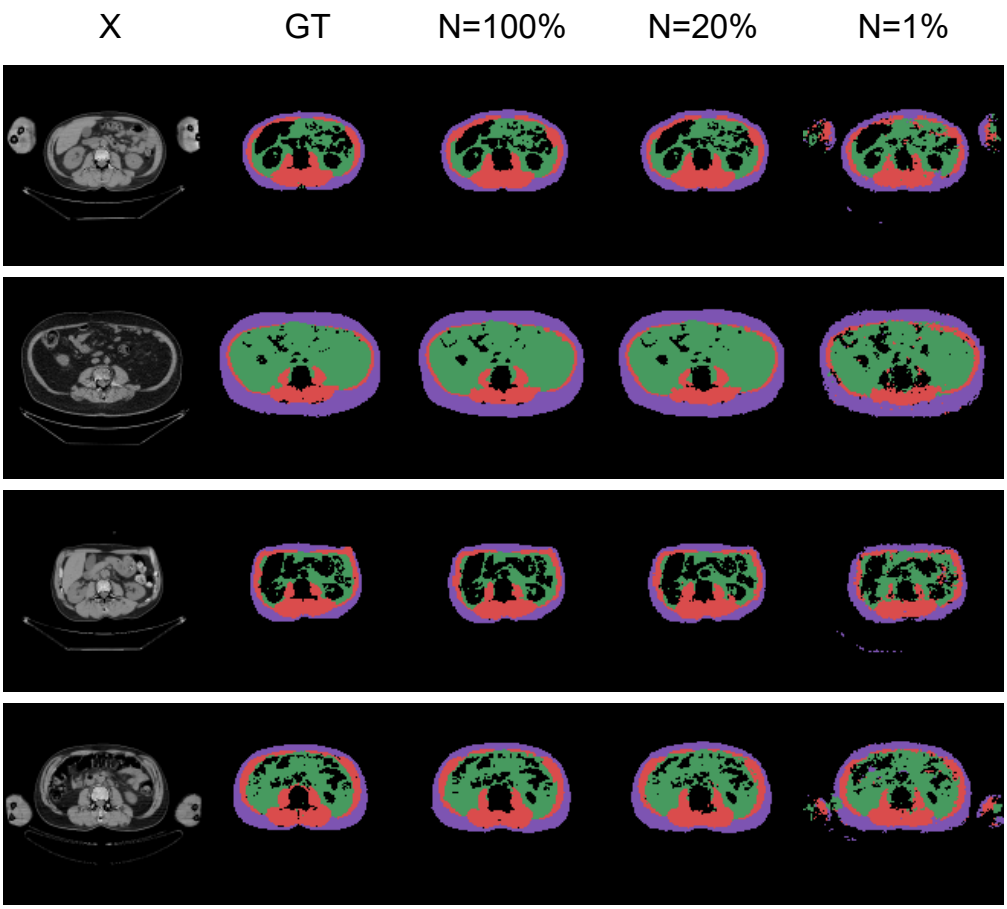


Fig. 21: Predicted sample from LSM on SSS test set, with different proportion of (x, y) available (100%, 20%, 1%)