MINIMALLY SUPERVISED MEDICAL SEMANTIC SEGMENTATION VIA UNSHARED RECONSTRUCTIONS AND SHARED FEATURE VECTORS

Anonymous authors
Paper under double-blind review

ABSTRACT

The research frontier in semi-supervised semantic image segmentation has recently shifted from the moderate supervision regimes towards minimal supervision, involving 1% to 5% labeled data. This is particularly relevant to medicine, where data annotation requires scarce expert clinical time. However, the extreme low-data setting introduces a unique set of mathematical and structural challenges that do not linearly scale down from their more moderate counterparts. In this work, we introduce a novel minimally supervised ($\sim$1% labeled samples) framework for semantic segmentation of medical images. Our model learns meaningful representations through image reconstruction in an encoder–decoder network architecture setting. However, unlike conventional methods that generate a unique latent embedding per image, our architecture utilizes a novel sub-module, named the *Vectorizer*, and outputs a small set of scalar coefficients. These coefficients serve the purpose of generating a linear combination of trainable feature vectors shared across the dataset. This allows the proposed model to rely on prior shape knowledge accumulated throughout training in order to reconstruct each class. We tested our framework on challenging and diverse medical image datasets, surpassing or being comparable with the minimally supervised state of the art.

1 INTRODUCTION

Semantic segmentation provides a pixel-level understanding of images and is used in a wide range of applications. Supervised deep learning approaches have delivered state-of-the-art accuracy, but come with major practical limitations such as labeling cost, annotation bias, and data scarcity, to name a few. The above especially holds in fields such as medical imaging, where, apart from the high-skill labor required to acquire and annotate each sample, privacy concerns magnify the difficulty of data availability. In order to overcome those challenges, researchers have shown interest in self-, semi-, and minimally supervised approaches (Wang et al., 2024).

Self-supervised learning relies on the idea that there is information in the image itself, regardless of whether it comes paired with a detailed label (Marks et al., 2025). The simplest form of neural network that aims to extract this information is the autoencoder, which compresses the image into a low-dimensional latent vector with an encoder, then uses that vector as input to a decoder that reconstructs the image (Bank et al., 2020). More recent adaptations aim at predicting a missing part of the image (Pathak et al., 2016), or at predicting missing patches after transformers entered computer vision (He et al., 2022). They have also produced remarkable results as foundation models (Caron et al., 2021). The case of medical images is more complicated, compared to natural ones, since the same objects, such as organs, bones and muscles, tend to appear in the same areas across images and might have very similar structure, especially tissues of the same type. Self-supervised methods struggle to separate such highly correlated structures without any supervision (Singh & Mishra, 2024).

Minimally supervised learning keeps a tiny labeled set and pairs it with a much larger unlabeled one, so that a supervised loss on the labels can be combined with a training signal computed on the rest of the data (Van Engelen & Hoos, 2020). In medical imaging, where even a handful of expert-annotated cases already represents a meaningful investment of clinical time, this labeled set can be pushed to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

a very small fraction of the data, on the order of 1% of the training set (Chen et al., 2021; Bai et al., 2023). At this level of scarcity, the labeled set is also unlikely to cover the full range of anatomical variation present in the rest of the data, and a model trained directly on it overfits easily (Tajbakhsh et al., 2020). Many semi-supervised methods address this by generating a training signal for the unlabeled data from a model that is itself trained on the same small labeled set, for example through a teacher network, the pseudo-labels of a second network, or an interpolated target. However the quality of that signal therefore depends on how well that model has been trained, which is limited at 1% supervision. For that reason, we believe that the labels are better used as an anchor, that tells the model which class is which, rather than as the main source of the training signal.

Medical imaging spans many modalities, including X-ray, Computed Tomography (CT) (Mazonakis & Damilakis, 2016), Magnetic Resonance Imaging (MRI) (Van Geuns et al., 1999), and ultrasound (Gillam & Marcoff, 2024), each suited to different organs, tissues, and clinical questions. Segmentation is a building block for many of the automated pipelines built on top of these modalities, supporting tasks such as organ volumetry, lesion tracking, and treatment planning (Litjens et al., 2017). At the same time, the volume of images produced by radiology departments has grown faster than the number of specialists available to read them (McDonald et al., 2015), so methods that reduce the need for manual annotation are valuable for every modality, not only for a specific one. Ultrasound is a particularly attractive target for such methods, even though it offers a lower spatial resolution and a smaller field of view than CT or MRI, as it is fast, portable, inexpensive (Wei et al., 2023) and does not expose the patient to radiation and it is the highest volume modality in routine use. Deep learning has already been shown to extract clinically useful measurements, such as right ventricular volume, directly from it (Bohoran et al., 2023; Kampaktsis et al., 2024). Our framework, described below, is not tied to any single modality or anatomy; we demonstrate it primarily on cardiac ultrasound, but also evaluate it on a broad range of other medical imaging tasks.

2 Related Work

Recent semi-supervised methods go beyond simple consistency regularization and address specific problems that appear when labeled data are scarce. VQ-Seg replaces the continuous feature dropout typically used for perturbation with a learned codebook: features are discretized into tokens, and perturbation is applied by shuffling the spatial arrangement of the token indices, which keeps every perturbed feature within the space of anatomically valid tokens (Yang et al., 2025). DyCON targets the class imbalance and boundary ambiguity typical of medical volumes with two losses, an uncertainty-aware consistency loss that keeps high-uncertainty voxels in the training signal instead of filtering them out, and a focal, entropy-aware contrastive loss that prioritizes hard positive and negative pairs at the pixel level (Assefa et al., 2025). ARCO addresses the same imbalance from a variance-reduction perspective, showing that naively sampled contrastive pairs let tail classes get conflated with the dominant background, and introducing stratified sampling schemes that provably reduce the variance of the resulting gradient estimator (You et al., 2023). MiDSS instead targets the domain shift that arises when the small labeled set and the much larger unlabeled pool come from different clinical centers, constructing an intermediate domain by pasting labeled and unlabeled content into each other and training with a symmetric guidance signal derived from the intermediate samples (Ma et al., 2024).

More closely related to our approach, reconstruction-based methods rely on the fact that similar objects appear similar across all images. Decomposing an image into object prototypes (Monnier et al., 2021) works by keeping a set of trainable object prototypes in image space; after each is passed through a trainable data augmentation, the resulting objects are composed to reconstruct the original image, and the segmentation is obtained by tracking which object came from which prototype. Because the augmentations are themselves trainable this offers some flexibility, but it retains the limitation that the reconstructed image must be built out of the prototypes' own appearance in image space, which is not always the case. Later work moved this decomposition into latent space instead (Wu et al., 2018): slot attention (Locatello et al., 2020) and IODINE (Greff et al., 2019) obtain a set of object-centric latent vectors through iterative refinement, and more recent variants such as Slot-VAE (Wang et al., 2023) and Semantic Vector Quantization (Wu et al., 2023) improve their generative and detection quality further.

In this paper, we make use of the idea of decomposing an image into per-class feature vectors, using a reconstruction loss as part of the training signal. This places our method between the two families described above, but with important differences from each:

- Unlike the semi-supervised methods, the reconstruction signal we rely on is computed identically on labeled and unlabeled images alike, and does not depend on a model trained on the small labeled set. As a result, it does not depend on the quality of a teacher's prediction, a pseudo-label or an interpolated target, all of which are only as reliable as the model that produced them.
- Unlike slot attention, IODINE, and prototype-based decomposition, our components are not unordered. Each *Vectorizer* is dedicated to a specific semantic class from the start, instead of being randomly initialized or iteratively refined. The small labeled set is used to bind each *Vectorizer* to its class.
- We deliberately limit the information that flows from the input image to the decoders, forcing the model to rely on prior knowledge accumulated across the dataset to reconstruct each class. This matters in real world degraded medical images, where a structure such as an organ or cavity may not be clearly visible in a given image, but is nonetheless likely to occupy the same general area as in the rest of the samples.

3 METHOD

3.1 THE ALGORITHM

On a high-level description, our model takes in an input image $\mathbf{x} \in [0,1]^{C\times H\times W}$ and, in a typical encoder fashion, produces a single vector a

$$a = Encoder(\mathbf{x}). \tag{1}$$

This output vector is then used as input for a sub-module called *Vectorizer*. There are n separate *Vectorizers*, each producing a component vector c_i that is passed as input to the decoders, to partially reconstruct and segment the image. We call n the number of components. Unlike a fully unsupervised setting, where the number of components is a design choice and the semantic classes have to be recovered from the components during a post-processing step, here n is equal to the number of semantic classes, background included, and component i is dedicated to class i from the start. While the idea of components is similar to the aforementioned slots (Locatello et al., 2020), there are key differences on how they are generated and on what they represent, so we will call them components to avoid any confusion. Inside the *Vectorizer* i, the output a of the encoder is downsampled through a typical neural network into a vector $s_i \in \mathbb{R}^d$, d standing for degrees of freedom

$$s_i = Vnn_i(a), \tag{2}$$

where Vnn_i is the neural network of *Vectorizer* i. Inside each *Vectorizer*, there are d learnable feature vectors $u_{ij} \in \mathbb{R}^e$, with e being the embedding dimension and j denoting the vector. The component vectors are obtained by the following equation

$$c_i = \sum_{j=0}^{d-1} s_{ij}\, u_{ij}, \tag{3}$$

where s_{ij} denotes each element of the vector s_i and is treated as a scalar value that is used to weight the corresponding feature vector u_{ij}. At the end of this process, there are n component vectors $c_{0,\dots,(n-1)} \in \mathbb{R}^e$. Each component vector is a linear combination of the trainable feature vectors of its own *Vectorizer*. These component vectors are then used as input to two decoders, one creating the reconstruction and one creating the segmentation of the corresponding component

$$Rec_i = Decoder_{rec}(c_i), \qquad Seg_i = Decoder_{seg}(c_i), \tag{4}$$

with $Rec_i \in [0,1]^{C\times H\times W}$ and $Seg_i \in \mathbb{R}^{H\times W}$. The final segmentation is the result of a softmax (Banerjee et al., 2020) operation, applied pixel-wise across all the component segmentations

$$P_i = \frac{e^{Seg_i}}{\sum_{j=0}^{n-1} e^{Seg_j}}, \tag{5}$$

so that $P_i(p)$ is the probability of pixel p belonging to class i. The complete framework is illustrated in Figure 1.

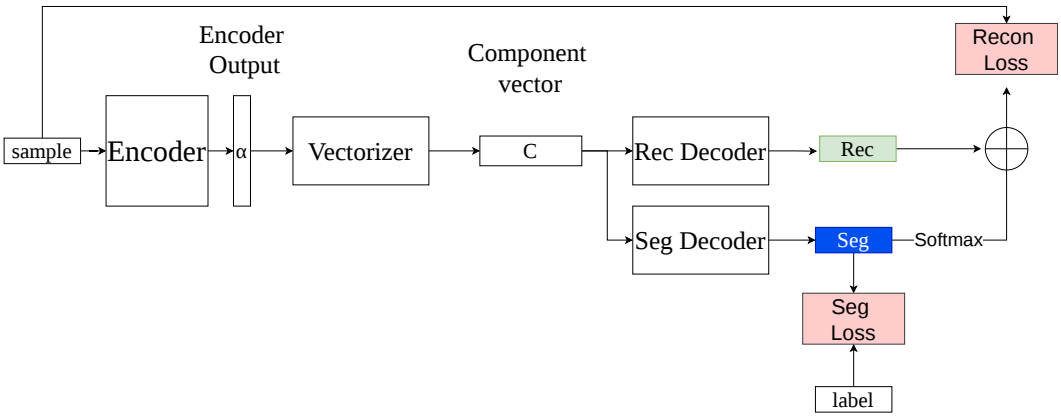


Figure 1: Complete framework overview, shown for a single component. The input sample $\mathbf{x}$ is passed to the encoder that produces a single output vector a. This vector is then used as input to the *Vectorizer*, that produces a component vector c. The component vector is passed to two separate decoders, a reconstruction decoder and a segmentation decoder, both shared across all components. The segmentations of all components are normalized using a softmax and are used to mask their corresponding reconstructions, which are compared against the sample (reconstruction loss). On labeled samples, the segmentation is also compared against the label (segmentation loss).

3.2 Encoder

Our goal was to make our framework as versatile as possible, allowing for modifications that fit to each problem it is applied to. The encoder can be any type of architecture, from a Convolutional Neural Network (CNN) to a transformer, or any type of architecture that takes an input image and produces a single output vector. As a matter of fact, any output could be used, such as feature maps, as long as the input layers of the *Vectorizers* are adapted accordingly. In our work, the encoder follows a residual network architecture, similar to the famous ResNet (He et al., 2016). It consists of a strided convolution followed by a sequence of downsampling stages, each halving the spatial resolution and followed by a residual double convolution block. The resulting feature map is flattened into the vector a, which is shared by all the *Vectorizers*.

3.3 Vectorizer

The main novelty of our framework is the use of a *Vectorizer* to derive the component vectors, shown in Figure 2. Reconstructing an input image is a very common method of self-supervised learning, but one has to keep in mind that it does not constitute the end goal, rather a means to it. From our experiments, we have found that from a certain point onwards, there is a tradeoff between image reconstruction quality and segmentation results. As medical images are often of low quality, objects may be obstructed or not visible. A model that derives all of its information from the input image would miss them. For that reason, limiting the information flow from the encoder to the decoders is essential for good segmentation quality. On the other hand, lowering the dimensionality of the latent vector in a traditional autoencoder would result in worse segmentation results, as the decoder would have very limited information to work with. The *Vectorizer* sub-module combines the best of both worlds. Its input is the high dimensional output of the encoder, which is passed through a narrow hidden layer and projected onto the coefficients $s_i \in \mathbb{R}^d$. The output of the *Vectorizer* is a weighted sum of its internal trainable feature vectors $u_{ij} \in \mathbb{R}^e$, multiplied by their corresponding element s_{ij}. The feature vectors are not a function of the input; they are shared across all the images of the dataset and are updated by every one of them, labeled or not. That way, a big portion of the information that reaches the decoders is derived from the shared feature vectors u_{ij} and not directly from the image. This enforces strong, trainable shape priors that are shared across all images, without us having to explicitly model them using domain specific knowledge. Each class has its own *Vectorizer*, and therefore its own set of feature vectors, that aims at reconstructing and segmenting a different part of the image. One may notice that the operation between s_i and the vectors $u_{i,0,\ldots,(d-1)}$ is practically

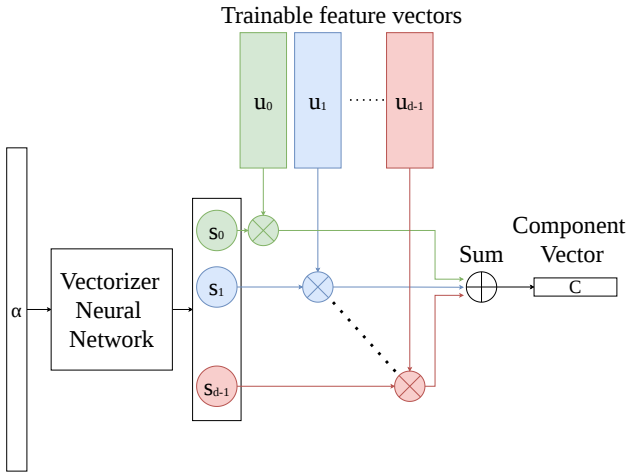


Figure 2: *Vectorizer* sub-module (the index i of the *Vectorizer* is omitted). Vector a is downsampled into vector s_i. Each element of that vector is used as a scalar, multiplying the corresponding feature vector u_{ij}. The component vector is the sum of those weighted vectors.

a matrix multiplication between vector s_i and the matrix that consists of all $u_{i,0,\ldots,(d-1)}$, stacked. That being said, we opted to describe the process using feature vectors, as they are more prevalent in the literature, instead of feature matrices.

3.4 Decoder

Similar to the encoder, the decoders can be anything that receives the output of a *Vectorizer* and produces an image. In our case, each decoder is a CNN that lifts the component vector to a small feature map with a transposed convolution, followed by a sequence of bilinear upsampling stages with residual double convolution blocks, and a final 1×1 convolution. The reconstruction decoder produces C channels, followed by a sigmoid, while the segmentation decoder produces a single channel of logits. Unlike the *Vectorizers*, there is a single pair of decoders for all components, but each component is inferred separately.

The only information about the input image that reaches the decoders is the component vector c_i. This means that not only the textures of each class are learnable, through the feature vectors, but their positions as well. This is a deliberate choice, as in medical images, organs and structures more or less occupy the same position in every image, relative to other objects. A decoder that cannot copy the input is forced to rely on this prior knowledge. The reconstruction and the segmentation are produced by two separate decoders that share only their input c_i.

3.5 Loss function

The training set consists of a small labeled set $\mathcal{D}_l$ of image and label pairs $(\mathbf{x}, \mathbf{y})$, with $\mathbf{y} \in \{0,1\}^{n \times H \times W}$ a one-hot label, and the complete set of training images $\mathcal{D}_u$. As the reconstruction does not require any labels, the labeled images are also part of $\mathcal{D}_u$. The loss formula is the following

$$loss = L_{seg} + \lambda(t)\, L_{rec}, \tag{6}$$

with L_{seg} computed on images of $\mathcal{D}_l$ and L_{rec} computed on images of $\mathcal{D}_u$.

Reconstruction loss. A straightforward way to use the reconstruction as a supervision signal is to sum the partial reconstructions, weighted by their segmentations, $Rec = \sum_i Rec_i * P_i$, and compare the result against the input image. As a result of the multiplication, the derivative of each segmentation depends on the reconstructions of all the components, which is suboptimal. In this work, we swap the order of the two operations: the error of each component is computed first and is

then weighted by the corresponding segmentation

$$L_{rec} = \frac{1}{HW} \sum_{p} \sum_{i=0}^{n-1} P_i(p) \left\| Rec_i(p) - \mathbf{x}(p) \right\|^2, \tag{7}$$

where p denotes a pixel, and the squared error is averaged over the C channels of the image. Although this change is small, it changes what the segmentation learns. When the reconstructions are blended before the comparison, the segmentation acts as a mixing ratio, and two wrong reconstructions can produce a perfect result by canceling each other out. For example, for a pixel with value 0.5 and two component reconstructions of 0.2 and 0.8, an even split of the segmentation yields zero blended error, while both reconstructions are wrong by 0.3. Squared errors, on the other hand, are non-negative and cannot cancel out. Moreover, L_{rec} is linear in P_i, and its derivative with respect to the segmentation of a component

$$\frac{\partial L_{rec}}{\partial P_i(p)} \propto \left\| Rec_i(p) - \mathbf{x}(p) \right\|^2 \tag{8}$$

depends only on the reconstruction of that same component. In other words, it measures how well component i reconstructs pixel p, and it does not vanish in the areas where the segmentation has already committed to a class. As the minimum of a linear function over the probability simplex lies on one of its vertices, the optimal segmentation is a hard assignment of every pixel to the component that reconstructs it best. For that reason, there is no need for an additional regularization term that penalizes overlap between component masks. Up to a constant, L_{rec} is the negative log-likelihood of a per-pixel Gaussian mixture with unit variance, where Rec_i are the means of the mixture components and P_i are their responsibilities, a formulation reminiscent of the Expectation-Maximization algorithm (Dempster et al., 1977) and its neural variants (Greff et al., 2017). The main difference is that our components are not unordered: each one is bound to a specific semantic class, through the segmentation loss.

Segmentation loss. L_{seg} is the Dice loss (Milletari et al., 2016) between the final segmentation P and the label $\mathbf{y}$, averaged over all classes, background included

$$L_{seg} = 1 - \frac{1}{n} \sum_{i=0}^{n-1} \frac{2 \sum_p P_i(p)\, y_i(p)}{\sum_p P_i(p) + \sum_p y_i(p)}. \tag{9}$$

At the level of supervision we operate on, some classes are represented by only a handful of labeled pixels. A pixel-wise loss, such as cross-entropy, is dominated by the classes that occupy the largest areas and provides almost no gradient to the smaller ones. The Dice loss, on the other hand, weights every class equally, regardless of its size. The role of the segmentation loss is not to be the primary source of the training signal, rather to act as an anchor, identifying which component corresponds to which class. The bulk of the training signal comes from L_{rec}, which is computed identically on every training image and does not depend on the labels.

Reconstruction warmup. The reconstruction weight $\lambda(t)$ increases linearly from 0 to its final value λ over the first T_w epochs, and remains constant afterwards

$$\lambda(t) = \lambda \cdot \min\left(1, \frac{t}{T_w}\right), \tag{10}$$

where t denotes the epoch. From our experiments, when both losses are applied from the beginning, a single component tends to dominate the reconstruction, and the segmentation follows it, with the remaining classes never being predicted. With the warmup, the model first learns where each class is located, from the few available labels, and then learns to reconstruct each class, from all the available images.

3.6 Training

At every training step, a batch of images is drawn from $\mathcal{D}_u$ and a batch of image and label pairs is drawn from $\mathcal{D}_l$. A random geometric transformation (rotation, translation, scaling and shearing) is applied to every image, and to its label, if one exists. Photometric perturbations (Gaussian noise,

Table 1: Segmentation results with 1% labeled training data. **boldface** is used to highlight the best performance, while underline text is denotes the second best performance.

	Ours		ARCO		MiDSS		DyCON		BCP		CPS		ICT		MC-Net+		URPC	
Dataset	Dice↑	HD↓	Dice↑	HD↓	Dice↑	HD↓	Dice↑	HD↓	Dice↑	HD↓	Dice↑	HD↓	Dice↑	HD↓	Dice↑	HD↓	Dice↑	HD↓
CAMUS	0.897	**8.9**	**0.909**	9.3	0.8	10.2	0.868	28.8	0.858	9.1	0.874	10.0	0.895	11.0	0.884	10.0	0.882	10.4
CardiacUDA	**0.797**	**15.4**	0.744	16.4	0.792	28.4	0.523	55.6	0.788	19.9	0.796	15.7	0.791	25.3	0.793	19.8	0.771	21.8
Drishti-GS	**0.698**	**21.2**	0.642	22.9	0.594	52.2	0.694	30.9	0.449	55.3	0.364	101.5	0.524	56.2	0.594	40.1	0.581	31.4
HC18	0.915	**13.2**	**0.954**	17.3	0.866	32.3	0.432	69.3	0.769	50.4	0.916	23.3	0.743	42.5	0.918	22.5	0.813	41.4
ISIC 2017	0.743	**38.4**	0.735	44.6	0.716	49.1	0.680	53.0	0.732	43.7	**0.754**	42.4	0.718	47.7	0.737	43.8	0.748	43.9
MSD Hippocampus	**0.712**	**59.7**	0.648	82.6	0.627	85.6	0.504	128.5	0.458	133.9	0.519	129.6	0.482	114.4	0.584	103.7	0.572	109.2
RIM-ONE DL	**0.559**	**44.7**	0.494	57.4	0.558	50.2	0.545	47.8	0.506	54.5	0.393	150.8	0.136	197.3	0.321	124.6	0.419	84.6

brightness and contrast changes) are then applied to the input of the model only. The target of the reconstruction is the geometrically transformed but otherwise clean image, so that the model does not learn to reproduce the noise, but rather the structures underneath it. The reconstruction loss is computed on the batch of $\mathcal{D}_u$, the segmentation loss on the batch of $\mathcal{D}_l$, and the whole network, including the feature vectors, is trained end-to-end with a single optimizer. At inference time, only the encoder, the *Vectorizers* and the segmentation decoder are used, and the predicted class of every pixel is the one with the highest P_i.

4 Experiments

Our main objective was to segment the chambers of the heart in four-chamber view echocardiography images. To expand the scope of our framework, we tested it on a variety of imaging modalities and tasks, such as Magnetic Resonance Imaging (MRI) and the brain, fetal ultrasound, retinal fundus photography and dermoscopy. More specifically, the datasets are the following: CAMUS (Leclerc et al., 2019) and CardiacUDA (Yang et al., 2023) for echocardiography, HC18 (van den Heuvel et al., 2018) for fetal ultrasound, MSD Hippocampus (Antonelli et al., 2022) for brain MRI, Drishti-GS (Sivaswamy et al., 2014) and RIM-ONE DL (Batista et al., 2020) for retinal fundus photography, and ISIC 2017 (Codella et al., 2018) for dermoscopy. For every dataset, we used the train, validation and test split provided by the dataset authors, when available; otherwise, a random 70/15/15 split was used. A 1% labeled fraction was used for all the experiments. Since the labeled pool is extremely limited, which samples happen to be labeled has a severe impact on the final result. For example, if a class is fully or partially obstructed in one of the few annotated samples, the model will tend to under-predict that class. For that reason, in all of our experiments, the labeled training samples were the same across all baselines. Additionally, the data loading pipeline was identical for all runs. We compare our framework against recent semi-supervised methods, ARCO (You et al., 2023), MiDSS (Ma et al., 2024) and DyCON (Assefa et al., 2025), as well as established ones, BCP (Bai et al., 2023), CPS (Chen et al., 2021), ICT (Verma et al., 2019), MC-Net+ (Wu et al., 2022) and URPC (Luo et al., 2021). The results are found in Table 1. All the baselines were run using the code provided by their authors.

4.1 Results

As seen in Table 1, our framework outperforms all the baselines on most of the datasets. The low HD values show that our predictions rarely contain regions far from the actual structure, which is expected. The shared feature vectors act as a shape prior and keep the segmentation consistent through samples. Figure 3 corroborates the framework's ability to segment classes that have unclear boundaries in specific samples. We can see that parts of the chamber walls are not visible in the original images, where the chambers blend in with the background, as often happens in the CardiacUDA dataset, but are present in the reconstructions. The reason behind this observation is the fact that the information is derived from the shared trainable feature vectors. As in most samples the walls that separate the chambers from the background are present, the *Vectorizer* that is responsible for the reconstruction of that part of the image has learned to reconstruct them, even if they are not visible in that specific sample.

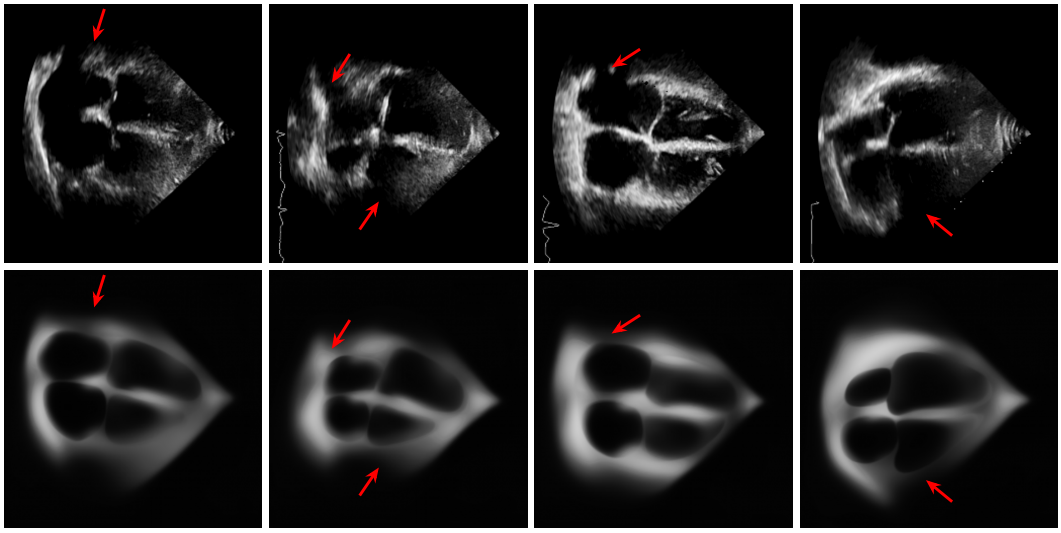


Figure 3: Original images (top) and their reconstructions (bottom) from the CardiacUDA dataset. The red arrows indicate chamber walls that are not visible in the original images, but are present in the reconstructions, as the reconstruction derives most of its information from the shared feature vectors.

5 Study Limitations and Future Work

An important thing to note about the framework is the fact that both the reconstruction and the segmentation decoder receive a single component vector, and no other information about the input image. This means that not only the different textures of each part of the image are learnable, through the feature vectors, but their positions as well. We have tested our framework on datasets where the location of each class varies, and the results are suboptimal. That being said, for medical imaging, that constraint is actually beneficial, as the organs and the structure of the body more or less remain the same in every human being, relative to other objects. For adaptations in other fields, such as natural images, the segmentation decoder should be given access to spatial information from the input image, for example through skip connections from the encoder.

AI use statement

In this work, we used generative AI tools to polish writing, write simple code to visualize and plot our results, aid with the download of datasets and the execution of the baseline code from the authors' GitHub repositories. This source code was not altered by the LLM, which was mainly used to convert the data from our own preprocessing pipeline into the format required by each repository. We have not used generative AI tools for writing our models' source code, drafting whole sections of the paper, synthetic data generation or proving mathematical claims. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

All the data used in this work are publicly available. Hence, there are no concerns about privacy or ethical issues.

Reproducibility statement

All source code is available in the zip folder in the supplementary materials. It will be publicly available on our github upon acceptance. The experiments were run using seed=1 and the hyperparameters in Table 2 in Appendix A.

REFERENCES

Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, et al. The medical segmentation decathlon. *Nature communications*, 13(1):4128, 2022.

Maregu Assefa, Muzammal Naseer, Iyyakutti Iyappan Ganapathi, Syed Sadaf Ali, Mohamed L. Seghier, and Naoufel Werghi. DyCON: Dynamic uncertainty-aware consistency and contrastive learning for semi-supervised medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Yunhao Bai, Duowen Chen, Qingli Li, Wei Shen, and Yan Wang. Bidirectional copy-paste for semi-supervised medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Kunal Banerjee, Rishi Raj Gupta, Karthik Vyas, Biswajit Mishra, et al. Exploring alternatives to softmax function. *arXiv preprint arXiv:2011.11538*, 2020.

Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *arXiv preprint arXiv:2003.05991*, 2020.

Francisco José Fumero Batista, Tinguaro Diaz-Aleman, Jose Sigut, Silvia Alayon, Rafael Arnay, and Denisse Angel-Pereira. Rim-one dl: A unified retinal image database for assessing glaucoma using deep learning. *Image Analysis and Stereology*, 39(3):161–167, 2020.

T. A. Bohoran, P. N. Kampaktsis, L. McLaughlin, J. Leb, S. Moustakidis, G. P. McCann, and A. Giannakidis. Right ventricular volume prediction by feature tokenizer transformer-based regression of 2D echocardiography small-scale tabular data. In *International Conference on Functional Imaging and Modeling of the Heart*, pp. 292–300. Springer, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pp. 168–172. IEEE, 2018.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Linda D. Gillam and Leo Marcoff. Echocardiography: Past, present, and future. *Circulation: Cardiovascular Imaging*, 17(4):e016517, 2024.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning (ICML)*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

P. N. Kampaktsis, T. A. Bohoran, M. Lebehn, L. McLaughlin, J. Leb, Z. Liu, S. Moustakidis, A. Siouras, A. Singh, R. T. Hahn, et al. An attention-based deep learning method for right ventricular quantification using 2D echocardiography: feasibility and accuracy. *Echocardiography*, 41(1):e15719, 2024.

Sarah Leclerc, Erik Smistad, Joao Pedrosa, Andreas Østvik, Frederic Cervenansky, Florian Espinosa, Torvald Espeland, Erik Andreas Rye Berg, Pierre-Marc Jodoin, Thomas Grenier, et al. Deep learning for segmentation using an open large-scale dataset in 2d echocardiography. *IEEE transactions on medical imaging*, 38(9):2198–2210, 2019.

Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A. W. M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42: 60–88, 2017.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11525–11538, 2020.

Xiangde Luo, Wenjun Liao, Jieneng Chen, Tao Song, Yinan Chen, Shichuan Zhang, Nianyong Chen, Guotai Wang, and Shaoting Zhang. Efficient semi-supervised gross target volume of nasopharyngeal carcinoma segmentation via uncertainty rectified pyramid consistency. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*. Springer, 2021.

Qinghe Ma, Jian Zhang, Lei Qi, Qian Yu, Yinghuan Shi, and Yang Gao. Constructing and exploring intermediate domains in mixed domain semi-supervised medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Markus Marks, Manuel Knott, Neehar Kondapaneni, Elijah Cole, Thijs Defraeye, Fernando Perez-Cruz, and Pietro Perona. A closer look at benchmarking self-supervised pre-training with image classification. *International Journal of Computer Vision*, 133(8):5013–5025, 2025.

M. Mazonakis and J. Damilakis. Computed tomography: What and how does it measure? *European Journal of Radiology*, 85(8):1499–1504, 2016.

Robert J. McDonald, Kara M. Schwartz, Laurence J. Eckel, Felix E. Diehn, Christopher H. Hunt, Brian J. Bartholmai, Bradley J. Erickson, and David F. Kallmes. The effects of changes in utilization and technological advancements of cross-sectional imaging on radiologist workload. *Academic Radiology*, 22(9):1191–1198, 2015.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, 2016.

Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry. Unsupervised layered image decomposition into object prototypes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Azad Singh and Deepak Mishra. CoBooM: Codebook guided bootstrapping for medical image representation learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 23–33. Springer, 2024.

Jayanthi Sivaswamy, SR Krishnadas, Gopal Datt Joshi, Madhulika Jain, and A Ujjwaft Syed Tabish. Drishti-gs: Retinal image dataset for optic nerve head (onh) segmentation. In *2014 IEEE 11th international symposium on biomedical imaging (ISBI)*, pp. 53–56. IEEE, 2014.

Nima Tajbakhsh, Laura Jeyaseelan, Qian Li, Jeffrey N. Chiang, Zhihao Wu, and Xiaowei Ding. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, 63:101693, 2020.

Thomas LA van den Heuvel, Dagmar de Bruijn, Chris L de Korte, and Bram van Ginneken. Automated measurement of fetal head circumference using 2d ultrasound images. *PloS one*, 13(8): e0200412, 2018.

Jesper E. Van Engelen and Holger H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

Robert-Jan M Van Geuns, Piotr A Wielopolski, Hein G de Bruin, Benno J Rensing, Peter MA Van Ooijen, Marc Hulshoff, Matthijs Oudkerk, and Pim J de Feyter. Basic principles of magnetic resonance imaging. *Progress in cardiovascular diseases*, 42(2):149–156, 1999.

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, pp. 3635–3641, 2019.

Mengzhu Wang, Jiao Li, Houcheng Su, Nan Yin, Liang Yang, and Shen Li. Graphcl: Graph-based clustering for semi-supervised medical image segmentation. *arXiv preprint arXiv:2411.13147*, 2024.

Yanbo Wang, Letao Liu, and Justin Dauwels. Slot-vae: Object-centric scene generation with slot attention. In *International Conference on Machine Learning*, pp. 36020–36035. PMLR, 2023.

C. Wei, M. Milligan, M. Lam, P. A. Heidenreich, and A. Sandhu. Variation in cost of echocardiography within and across United States hospitals. *Journal of the American Society of Echocardiography*, 36(6):569–577, 2023.

Yi-Fu Wu, Minseung Lee, and Sungjin Ahn. Object-centric semantic vector quantization. In *UniReps: the First Workshop on Unifying Representations in Neural Models*, 2023.

Yicheng Wu, Zongyuan Ge, Donghao Zhang, Minfeng Xu, Lei Zhang, Yong Xia, and Jianfei Cai. Mutual consistency learning for semi-supervised medical image segmentation. *Medical Image Analysis*, 2022.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Jiewen Yang, Xinpeng Ding, Ziyang Zheng, Xiaowei Xu, and Xiaomeng Li. Graphecho: Graph-driven unsupervised domain adaptation for echocardiogram video segmentation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11844–11853. IEEE, 2023.

Sicheng Yang, Zhaohu Xing, and Lei Zhu. VQ-Seg: Vector-quantized token perturbation for semi-supervised medical image segmentation. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Chenyu You, Weicheng Dai, Yifei Min, Fenglin Liu, David Clifton, S Kevin Zhou, Lawrence Staib, and James Duncan. Rethinking semi-supervised medical image segmentation: A variance-reduction perspective. *Advances in neural information processing systems*, 36:9984–10021, 2023.

A Appendix

Table 2: Hyperparameters of our method. One setting is used for every dataset.

Hyperparameter	Value
Architecture	
Input size	256×256
Encoder channels	16, 32, 64, 128, 256, 512 (6 stages)
Decoder channels	1024, 512, 256, 128, 64, 32, 16 (7 stages)
Decoder first kernel size	4
Class vector (bottleneck) size	256
Prototype vectors per class	64
Vectorizer MLP hidden width	128
Reconstruction / segmentation decoders	separate (no shared stages)
Normalization	BatchNorm
Losses	
Reconstruction loss	$\sum_k p_k (r_k - x)^2$ (square-then-blend)
Reconstruction weight	20
Reconstruction warm-up	linear from 0, 30 epochs
Segmentation loss	soft Dice (background included)
Optimization	
Optimizer	Adam, lr 10^{-4}, no weight decay, no schedule
Batch size	16 labeled + 16 unlabeled
Iterations	10,000
Mixed precision	yes
Augmentation	
Affine (image + label)	rotation $\pm 10^\circ$, translation $\pm 10\%$, scale 0.9–1.1, shear $\pm 15^\circ$
Input only	Gaussian noise $\sigma = 0.05$, brightness $\pm 20\%$, contrast $\pm 20\%$