Neural Networks 201 (2026) 108925

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

Biologically-inspired semi-supervised semantic segmentation for biomedical imaging

Luca Ciampi [1,*], Gabriele Lagani [1,*], Giuseppe Amato, Fabrizio Falchi

ISTI-CNR, Pisa, Italy

ARTICLE INFO

Keywords:
Hebbian learning
Bio-inspired computer vision
Semi-supervised learning
Semantic segmentation
Biomedical imaging
Human-inspired computer vision

ABSTRACT

We propose a novel bio-inspired two-stage semi-supervised learning approach for training semantic segmentation models based on downsampling-upsampling architectures. The first stage does not use backpropagation. Rather, it exploits the Hebbian principle *"fire together, wire together"* as a local learning rule for updating the weights of both convolutional and transpose-convolutional layers, allowing unsupervised discovery of data features. In the second stage, the model is fine-tuned with standard backpropagation on a small subset of labeled data. We evaluate our methodology through experiments conducted on several widely used biomedical datasets, deeming that this domain is paramount in computer vision and is notably impacted by data scarcity. Results show that our proposed method outperforms SOTA approaches across different levels of label availability. Furthermore, we show that using our unsupervised stage to initialize the SOTA approaches leads to performance improvements. The code to replicate our experiments can be found at https://tinyurl.com/hebbian-semantic-segmentation.

1. Introduction

Semantic segmentation in biomedical images assigns pixel-level class labels to structures like cells, tumors, and lesions, playing a key role in automating computer-aided diagnoses. Recent data-driven deep-learning approaches using CNNs and Transformers have shown excellent results (Cao et al., 2023; Chen et al., 2017; Hatamizadeh et al., 2022; Long et al., 2015; Milletari et al., 2016; Ronneberger et al., 2015; Xie et al., 2021). However, these methods demand extensive annotated data for supervised backpropagation-based training, which restricts their large-scale application despite their significance in computer vision (Basak et al., 2022; Çiçek et al., 2016; Isensee et al., 2020; Karimi et al., 2023; Valanarasu et al., 2021, 2022).

Nevertheless, integration between biological mechanisms and deep learning (BIDL) seems a promising direction, tackling not only the substantial demand for data but also allowing complex neural computation with extreme energy efficiency (Schuman et al., 2022; Shrestha et al., 2022). In particular, the so-called Hebbian principle represents a simple local learning rule that closely mimics the synaptic adaptations of brain mechanisms (Gerstner & Kistler, 2002; Haykin, 2009; Lagani et al., 2023), offering an appealing and still relatively unexplored solution for extracting data features without relying on supervised backpropagation and on the availability of large labeled datasets. Different Hebbian learning strategies involving Soft-Winner-Takes-All (SWTA) (Krotov & Hopfield, 2019; Lagani et al., 2022a; Moraitis et al., 2022), or Principal Component Analysis (HPCA) (Bahroun & Soltoggio, 2017; Lagani et al., 2021; Pehlevan et al., 2015), allow a group of neurons to discover unsupervised features from a set of data such as clusters or principal components, and can be used to fulfill unsupervised and semi-supervised learning strategies coming to the rescue to mitigate data scarcity issues.

In this work, we introduce a two-stage semi-supervised biologically-inspired learning approach for semantic segmentation in biomedical images (Fig. 1). In the first stage, we exploit a set of unlabeled data to perform unsupervised pretraining of standard semantic segmentation models based on downsampling-upsampling architectures, using the Hebbian learning principle implemented with different strategies. In the second step, we use these weights to initialize the model and fine-tune it through standard backpropagation supervised training on a few labeled data samples. Specifically, during the unsupervised round, we employ Hebbian learning rules in standard convolutional layers in the downsampling path and—for the first time—for transpose-convolutional (T-Conv) layers used for upsampling the feature maps.

We perform an experimental assessment of our methodology over three public datasets widely used in the context of medical 2D image segmentation (Basak et al., 2022; Karimi et al., 2023; Valanarasu et al., 2021, 2022) showcasing different imaging modalities and segmentation tasks, i.e., GlaS (Sirinukunwattana et al., 2017) for

* Corresponding authors.
E-mail addresses: luca.ciampi@isti.cnr.it (L. Ciampi), gabriele.lagani@isti.cnr.it (G. Lagani).
[1] They contribute equally to this work.

https://doi.org/10.1016/j.neunet.2026.108925
Received 18 March 2025; Received in revised form 21 January 2026; Accepted 30 March 2026
Available online 1 April 2026

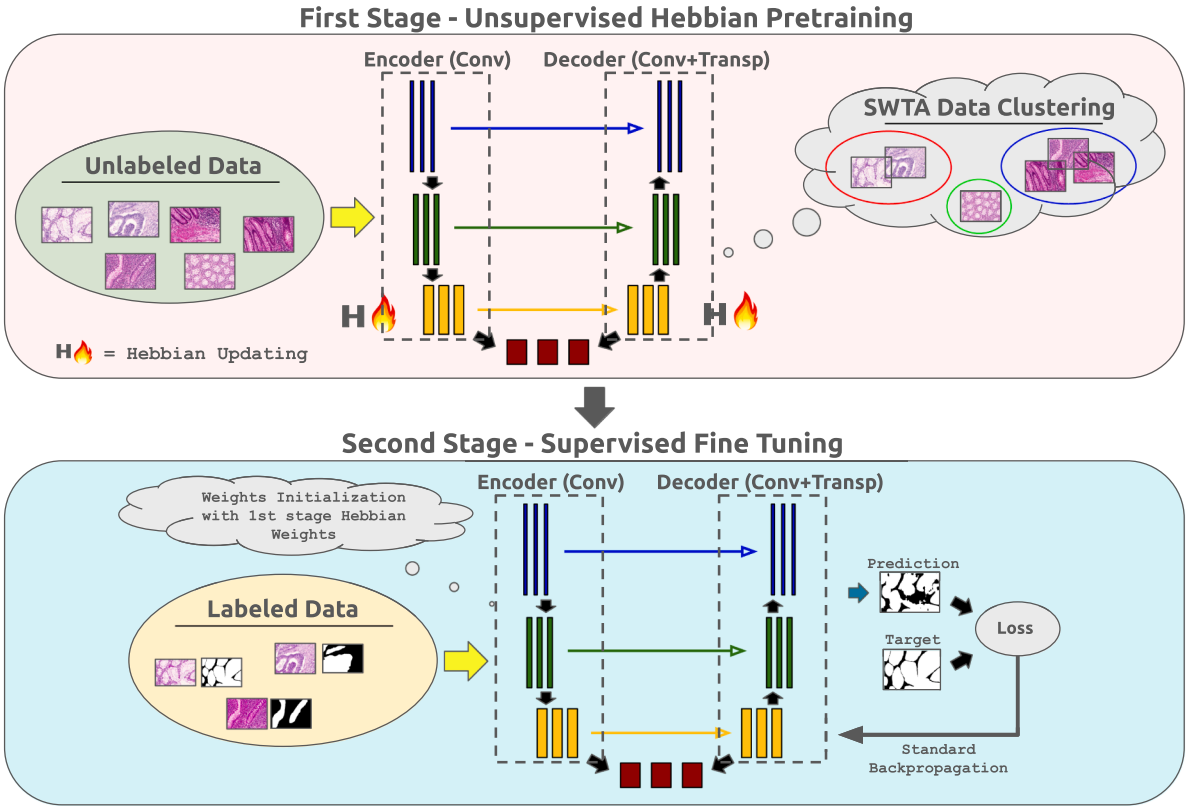


Fig. 1. Our semi-supervised, bio-inspired approach for semantic segmentation models with downsampling-upsampling architectures. The method unfolds in two stages. First, we employ Hebbian learning for unsupervised pre-training on a large set of unlabeled data, enabling the model to autonomously identify features like cluster centroids. Notably, we derive new Hebbian learning rules for the transpose-convolution layers in the upsampling path. In the second stage, we apply supervised backpropagation to fine-tune the model using a small labeled dataset.

colorectal cancer segmentation in Hematoxylin and Eosin (H&E) stained histological images, PH2 (Mendonça et al., 2013) for skin lesion segmentation in dermoscopic images, and HMEPS (Mazziotti et al., 2021) for eyes pupil segmentation in grayscale images. Moreover, we also conduct an analysis using the LA (Xiong et al., 2021) dataset for left atrial segmentation in MRIs, thus extending our method to volumetric images. We compare our approach against several SOTA single-stage semi-supervised approaches relying on pseudo-labeling and consistency training (Chen et al., 2021; Luo et al., 2021, 2022; Ouali et al., 2020; Vu et al., 2019; Yu et al., 2019), as well as some other two-stage pipelines exploiting Variational Auto-Encoders (VAEs) (Kingma & Welling, 2014) and different paradigms of Self-Supervised Learning (SSL) (Ouyang et al., 2020; Perez-Garcia et al., 2025) for the unsupervised step, alongside a bio-inspired model based on spiking neural networks (Kim et al., 2022). The outcomes demonstrate that our approach can substantially improve performance, considering various label scarcity conditions. Furthermore, we show that using our unsupervised Hebbian stage to initialize the SOTA single-stage approaches yields further improvements.

Concretely, our contributions are the following:

- We propose a novel semi-supervised two-stage pipeline for semantic segmentation, where a first unsupervised stage relying on the bio-inspired Hebbian principle is followed by a second supervised step that fine-tunes the model on a few labeled data samples.
- We validate our methodology on several public biomedical imaging benchmarks, demonstrating that our technique can substantially improve performance compared to SOTA methods across different degrees of label availability.
- We conduct a further experimental analysis by initializing existing SOTA semi-supervised approaches with our Hebbian unsupervised pre-training, showing performance improvements.

This work extends our previous workshop paper (Ciampi et al., 2024) that introduced the early-stage methodology and preliminary results. We significantly expand upon it in several directions by (i) formulating more advanced Hebbian learning rules specifically designed for T-Conv layers; (ii) conducting experiments on additional datasets and comparing our approach against a broader range of state-of-the-art methods; (iii) leveraging our unsupervised stage to initialize SOTA approaches, demonstrating performance improvements; and (iv) providing extensive ablation studies, as well as additional background and analysis, to offer a more comprehensive understanding and validation of the proposed approach.

We organize the rest of the paper as follows. In Section 2, we review related work. Section 3 provides a brief background on Hebbian learning. In Section 4, we describe our methodology, while Section 5 presents the experiments, discusses the results, and includes ablation studies to validate our approach. Finally, Section 6 concludes the paper. Additionally, Appendices A, B, C, D, and E provide further details on the background of Hebbian learning, Hebbian feature vectors latent space visualization, implementation details, additional experiments, and additional qualitative results, respectively.

2. Related works

2.1. Deep learning bio-inspired approaches

Recent Deep Learning (DL) models have achieved remarkable success across various tasks (Ciampi et al., 2022; Devlin et al., 2019; Dosovit-

skiy et al., 2020). However, key challenges remain, including the substantial demand for data (Roh et al., 2021) and energy (Badar et al., 2021). In contrast, biological intelligence appears to overcome these limitations (Javed et al., 2010; Lake & Piantadosi, 2020), suggesting that a tighter integration between biological mechanisms and DL (Bio-Inspired DL - BIDL) could be a promising direction (Hassabis et al., 2017; Lake et al., 2017). BIDL encompasses two complementary dimensions: the architecture and the learning algorithm. Regarding architectures, Spiking Neural Networks (SNNs) (Göltz et al., 2021; Lee et al., 2020; Wu et al., 2019; Zhou et al., 2021) model neural computation in a way that more closely resembles biological neurons. Unlike conventional architectures such as Convolutional Neural Networks (CNNs), which rely on continuous-valued activations and synchronous processing, SNNs operate using discrete spike events and asynchronous dynamics, enabling temporal coding and event-driven computation. In particular, neuromorphic hardware implementations based on SNNs offer complex neural computation with exceptional energy efficiency (Schuman et al., 2022; Shrestha et al., 2022; Wang et al., 2022), and recently SNNs have been successfully applied for computer vision tasks such as semantic segmentation (Kim et al., 2022; Lei et al., 2025; Patel et al., 2021; Yue et al., 2023). The other crucial aspect of BIDL pertains to learning algorithms. Conventional DL models rely on backpropagation. SNNs are fundamentally incompatible with backpropagation due to the non-differentiable nature of spike-based communication; for this reason, surrogate gradient methods are often employed. However, neither backpropagation nor surrogate gradient methods are biologically plausible and, consequently, alternative learning strategies inspired by biological synaptic plasticity have gained traction. Specifically, Hebbian learning (Gerstner & Kistler, 2002; Haykin, 2009; Journé et al., 2023; Lagani et al., 2021, 2022a) provides a biologically plausible alternative for learning in conventional DL systems (Bahroun & Soltoggio, 2017; Ciampi et al., 2024; Journé et al., 2023; Krotov & Hopfield, 2019; Lagani et al., 2022a,b; Moraitis et al., 2022) and its computational efficiency has already been analyzed in prior works (Gupta et al., 2022; Lagani et al., 2024, 2022b). In spiking architectures, a related mechanism is Spike-Timing Dependent Plasticity (STDP) (Bi & Poo, 1998), which updates synaptic weights based on the relative timing of spikes. In this work, we focus on the learning algorithm aspect and specifically investigate Hebbian learning applied to conventional neural architectures for semantic segmentation.

2.2. Semi-supervised semantic segmentation

The widespread adoption of deep learning has led to the extensive utilization of CNNs and Transformers in the realm of semantic segmentation, such as FCN (Long et al., 2015), SegNet (Badrinarayanan et al., 2017), and DeepLabV3 (Chen et al., 2017). Concerning biomedical images, UNet-like architectures have emerged to be the most efficient and performing ones (Cao et al., 2023; Çiçek et al., 2016; Hatamizadeh et al., 2022; Isensee et al., 2020; Milletari et al., 2016). To address the primary limitation of these approaches—namely, the limited availability of labeled data—semi-supervised techniques offer a promising solution. This paradigm aims to extract knowledge from a vast pool of unlabeled data in combination with supervised learning on a few labeled samples (Bengio et al., 2006; Larochelle et al., 2009).

Single-stage approaches. Dominant semi-supervised strategies include single-stage approaches categorized as (i) pseudo-labeling, where the model computes an auxiliary loss generating pseudo-targets associated with the unlabeled data, and (ii) consistency training, where the model is fed with different perturbed versions of a given input and is enforced to generate similar predictions associated to them, constructing an additional loss satisfying this consistency criterion (Chen et al., 2021; Luo et al., 2021, 2022; Ouali et al., 2020; Vu et al., 2019; Yu et al., 2019). ADVENT (Vu et al., 2019) is a pseudo-labeling approach that uses predictions of the network itself as pseudo-labels for unlabeled samples, minimizing the entropy of the output probability distribution. Uncertainty-Aware Mean Teacher (UAMT) (Yu et al., 2019) follows a student-teacher architecture in which the student model learns progressively from the teacher model's reliable predictions. The teacher model not only generates target outputs but also assesses the uncertainty of each prediction using Monte Carlo sampling. Cross-Consistency Training (CCT) (Ouali et al., 2020) is a consistency-based method that generates different predictions by applying various perturbations to the latent representation generated by the model from a given unlabeled input. A loss term encourages the model to align predictions from different perturbations. A more recent consistency-based approach is Uncertainty Rectified Pyramid Consistency (URPC) (Luo et al., 2022) which learns from unlabeled data by minimizing the discrepancy between a set of segmentation predictions at different scales. Conversely, Cross Pseudo Supervision (CPS) (Chen et al., 2021) enforces consistency between two segmentation networks with different initializations for the same input image. Each network generates a pseudo-one-hot label map, which is then used to supervise the other network through a standard cross-entropy loss. Instead, the authors in (Luo et al., 2021) proposed a Dual-Task Consistency (DTC) framework that jointly predicts a pixel-wise segmentation map and a geometry-aware level set representation of the target.

Two-stage approaches. An alternative semi-supervised approach involves an initial unsupervised training phase on unlabeled data, followed by fine-tuning on labeled samples (Bengio et al., 2006; Kingma et al., 2014; Larochelle et al., 2009; Zhang et al., 2016). Specifically, during the second stage, a semantic segmentation branch is trained on top of the features learned in the unsupervised phase. Techniques commonly used during the unsupervised phase include Variational Autoencoders (VAEs) (Kingma & Welling, 2014) and different paradigms of Self-Supervised Learning (SSL) (Ouyang et al., 2020; Perez-Garcia et al., 2025). Belonging to the latter category, (Ouyang et al., 2020) proposes a segmentation framework for medical imaging that relies on superpixel-based pseudo-labels. On the other hand, RAD-DINO (Perez-Garcia et al., 2025), along with other works such as UNI (Chen et al., 2024) and Prov-GigaPath (Xu et al., 2024), exploits the generalization capabilities of DINO-based (Caron et al., 2021) feature extractors to achieve state-of-the-art performance across a variety of downstream tasks, positioning these models as emerging foundation models in the biomedical domain. Similarly, Hebbian learning can also be employed in a two-stage fashion, where unsupervised pretraining based on biologically inspired local learning rules precedes task-specific fine-tuning. While Hebbian learning-based pretraining has previously been applied to image classification (Lagani et al., 2021), the use of Hebbian rules for weight updates in T-Conv layers has been largely unexplored and was only introduced in a preliminary form in our previous paper (Ciampi et al., 2024). Compared to traditional backpropagation-based approaches—such as those previously mentioned, including self-supervised learning, contrastive learning, pseudo-labeling, VAE-based, or DINO-based methods—Hebbian pretraining introduces a distinctive inductive bias rooted in biologically plausible mechanisms. Specifically, Hebbian synaptic dynamics enable layer-wise local weight updates, allowing each layer to learn independently of global error signals. This local learning paradigm fosters the emergence of structured representations aligned with clustering and pattern discovery principles. These properties make Hebbian learning particularly suitable for semantic segmentation, where spatial coherence and pixel-level class consistency are essential.

3. Background

This section provides a brief background about Hebbian learning rules for training artificial networks. More details can be found in Appendix A.

Neuroscientists formulated the plasticity occurring in synapses connecting different brain neurons proposing the Hebbian principle *"fire together, wire together"*. Essentially, it is grounded on the biological observation that neurons tend to strengthen their connections with other

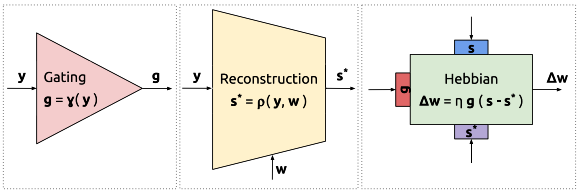


Fig. 2. Building blocks for Hebbian learning. Hebbian updates are computed from the difference between a target signal **s** and a reconstructed signal **s** $*$, and a gating signal **g** which modulates the update steps **g**. Gating and reconstruction signals are, in turn, derived from outputs and weights through the respective blocks.

neurons when their activities are correlated, or to weaken their couplings otherwise (Gerstner & Kistler, 2002). Mathematically, this behavior can be modeled by a simple learning rule implemented as follows (Haykin, 2009):

$$\Delta w_{i,j} = \eta\, y_j\, (x_i - w_{i,j}), \tag{1}$$

where x_i is the input stimulus delivered from neuron i to neuron j, y_j is the output of neuron j, $w_{i,j}$ are the weights connecting a neuron i and a neuron j, $\Delta w_{i,j}$ is the weight update computed by the learning rule, and η is the learning rate. The intuition behind this learning rule is that if a neuron is exposed to a set of inputs, the weight vector moves toward the centroid of the cluster formed by those inputs and the neuron becomes a detector of such a pattern (Grossberg, 1991; Haykin, 2009; Lagani et al., 2022a; Rumelhart & Zipser, 1985). Different formulations of this rule led to the Soft-Winner-Takes-All (SWTA) and the Hebbian Principal Component Analysis (HPCA) techniques.

Soft-Winner-Takes-All (SWTA). The authors of Grossberg (1991), Rumelhart and Zipser (1985) proposed the idea of *competitive learning*, to allow a set of neurons to specialize on different patterns: this is achieved by assigning higher weight updates to those neurons that best match the current input. Specifically, recent work (Lagani et al., 2021; Moraitis et al., 2022) demonstrated the effectiveness of Soft-Winner-Takes-All (SWTA) learning, i.e., a form of soft competition where neurons take update steps proportional to the softmax of their outputs:

$$\Delta w_{i,j} = \eta\, \mathrm{softmax}(y_1, y_2, \ldots)_j\, (x_i - w_{i,j}). \tag{2}$$

where $\mathrm{softmax}(y_1, y_2, \ldots)_j = \frac{e^{y_j/t}}{\sum_k e^{y_k/t}}$ and t is the temperature hyperparameter to tune the sharpness of the softmax operation ($t \to 0$ is equivalent to single WTA competition, while $t \to \inf$ means no competition).

Hebbian Principal Component Analysis (HPCA). The authors of Becker and Plumbley (1996), Karhunen and Joutsensalo (1995), Pehlevan et al. (2015), Sanger (1989) modeled lateral connectivity patterns observed in the brain (Luo, 2021; Pehlevan et al., 2015) and, from this model, they derived a new learning rule where contributions from neighboring neurons appear in the weight update of each neuron:

$$\Delta w_{i,j} = \eta\, y_j\, (x_i - \sum_{k=1}^{j} y_k w_{i,k}). \tag{3}$$

This formulation can be shown to be equivalent to Principal Component Analysis (PCA) (Becker & Plumbley, 1996; Karhunen & Joutsensalo, 1995; Sanger, 1989): by following this learning rule, the weight vectors of different neurons align towards the directions of the principal components of the data, in an efficient and biologically plausible fashion.

4. Method

4.1. Problem formulation

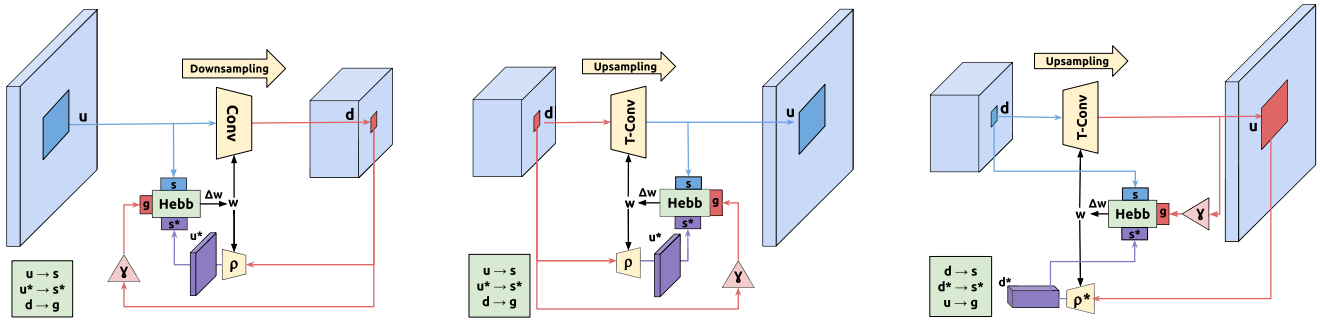
We assume to have a dataset $\mathcal{X} = \mathcal{X}_L \cup \mathcal{X}_U$, where $\mathcal{X}_L = \{(x_i, y_i)\}_{i=1}^{N_L}$ is the labeled subset including images x_i and corresponding pixel-wise labels y_i, and $\mathcal{X}_U = \{(x_j)\}_{j=1}^{N_U}$ is the unlabeled subset containing only images x_j. We also assume to be in typical semi-supervised settings, i.e., $N_L \ll N_U$ and $\mathcal{X}_L \cap \mathcal{X}_U = \emptyset$. Furthermore, we define an r*%-regime* as $\frac{r}{100}|\mathcal{X}_L|$, outlining several levels of label availability from $\mathcal{X}_L$ for the supervised training. We formulate a semi-supervised learning pipeline for semantic segmentation based on two steps (Fig. 1). In the first stage, unsupervised pre-training is performed using Hebbian learning algorithms on $\mathcal{X}_U$. Then, in the second stage, the model is fine-tuned via standard backpropagation on $\mathcal{X}_L$, across different label regimes r, using the standard cross-entropy loss.

4.2. Hebbian learning for T-Conv layers

Neural networks for semantic segmentation are usually characterized by encoder-decoder architectures, where a downsampling path responsible for computing image features is followed by an upsampling path that restores the spatial dimensions to make predictions pixel-wise (Badrinarayanan et al., 2017; Chen et al., 2017; Long et al., 2015). Specifically, UNet-like architectures (Ronneberger et al., 2015) have emerged to be the most efficient and performing models for semantic segmentation in biomedical images (Cao et al., 2023; Çiçek et al., 2016; Hatamizadeh et al., 2022; Isensee et al., 2020; Milletari et al., 2016). Here, convolutional layers make up the encoding stage, while the decoding step usually relies on T-Conv layers (Long et al., 2015) which are in general preferable compared to the combination of convolutional layers and interpolation (Dumoulin & Visin, 2016). As shown in Section 3, Hebbian learning strategies enable the unsupervised discovery of data features such as cluster centroids, and they have successfully been adopted in convolutional and fully-connected layers (Journé et al., 2022; Lagani et al., 2024, 2022b; Moraitis et al., 2022). However, a Hebbian theory for T-Conv layers is lacking, (it was only introduced in a preliminary form in our previous paper (Ciampi et al., 2024)). In this section, we illustrate our contributions to fill this gap.

Building blocks of Hebbian learning. For convenience, we decompose the Hebbian learning rules previously described in Eq. (2) and Eq. (3) into three basic building blocks, depicted in Fig. 2. The first block is a *gating* function $\gamma(\mathbf{y})$, which provides a factor to modulate the size of the update steps based on the neurons' outputs. It corresponds to $\gamma(\mathbf{y}) = \mathrm{softmax}(\mathbf{y})$ (the softmax function) for SWTA, and $\gamma(\mathbf{y}) = \mathbf{y}$ (the identity function) for HPCA, respectively. The second block is a *reconstruction* function $\rho(\mathbf{y}, \mathbf{w})$, which aims at reconstructing the neurons' input, given their activations and weights. It performs the following computations: first, the activation feature map undergoes a transformation that depends on whether we are performing SWTA or HPCA; then, a matrix multiplication of the result by the transposed weight matrix is performed. Concerning the first step, focusing on neuron j, the transformation corresponds to setting the j-th channel to 1 and the others to 0, in the case of SWTA, or to the identity for the first j channels and 0 for the others, in the case of HPCA. This leads to $\rho(\mathbf{y}, \mathbf{w})_{i,j} = w_{i,j}$ for SWTA, and $\rho(\mathbf{y}, \mathbf{w})_{i,j} = \sum_{k=1}^{j} y_k w_{i,k}$ for HPCA, respectively (with indexes i and j running over all inputs and outputs). The third block is the Hebbian plasticity function, which computes a weight update, given the following three signals as inputs: (i) a *target* signal (corresponding to the variable x_i), from which we want to discover patterns; (ii) a *reconstruction* signal, derived from the internal representation of the neural layer through the reconstruction block; (iii) a *gate*, derived from the neurons' activations through the gating block, that modulates the length of the weight update step.

The weight update follows the direction of the difference between the target and reconstructed signals. Calling **s** the target signal, **s** $*$ the



(a) Hebbian update computation in a standard convolutional layer, mapping an upsampled feature map to a downsampled one. A patch from the upsampled map serves as the target signal, while the gate and reconstruction signals are derived from a patch of the downsampled map via their respective blocks.

(b) Naive extension of Hebbian update computation to T-Conv layers: as before, a patch from the upsampled map serves as the target signal, while the gate and reconstruction signals are derived from a patch of the downsampled map via their respective blocks.

(c) Our formulation of Hebbian update computation for T-Conv layers: a patch from the downsampled map serves as the target signal, while the gate and reconstruction signals are derived from a patch of the upsampled map via a gating block and a custom-designed reconstruction block.

Fig. 3. Hebbian learning in convolutional and transpose-convolutional layers.

reconstruction, and **g** the gate, a general Hebbian learning rule can be written as:

$$\Delta w_{i,j} = \eta\, g_j\, (s_i - s^*_{i,j}). \tag{4}$$

In this newly introduced notation, s_i corresponds to x_i, s^*_i to $\rho(\mathbf{y}, \mathbf{w})_i$, and g_j to $\gamma(\mathbf{y})_j$, respectively.

Patch-wise Hebbian learning and the shape mismatch problem. Let us instantiate Eq. (4) in the specific case of standard convolutional layers. Let us refer to the feature map before the convolution as the *upsampled* feature map and to the feature map after the convolution as the *downsampled* feature map. In the convolutional case, Hebbian updates are obtained by processing the input patch-wise, as shown in Fig. 3a. In particular, the target signal corresponds to patches from the upsampled feature map, while reconstruction and gating terms are obtained from patches of the downsampled feature map through the respective blocks.

However, for T-Conv layers, we have a downsampled feature map before the layer and an upsampled feature map afterward. Simply applying the Hebbian learning rules in this scenario is not possible due to a *shape mismatch* problem in the reconstruction block: the latter block transforms downsampled feature maps to upsampled feature maps, but this is not compatible with T-Conv layers, where we need instead to transform an upsampled feature map into a downsampled reconstruction. In order to solve this issue, we identified two possible strategies, which are shown in Fig. 3b and Fig. 3c and described in the following.

First strategy: SWTA-S, HPCA-S. The first strategy is to exchange the role of downsampled and upsampled feature maps in the learning equations so that the same building blocks of Hebbian learning for standard convolutional layers can be reused in a different fashion. Specifically, as before, we use patches from the upsampled feature map as the target signal, while reconstruction and gating terms are computed from the corresponding patches in the downsampled feature map through the respective blocks. This methodology for applying Hebbian learning in T-Conv layers is straightforward because it allows us to simply reuse the same building blocks in Fig. 2 in these new settings. For this reason, we call the resulting learning methodologies following this *Straightfarward* extension as SWTA-S and HPCA-S.

Second strategy: SWTA-TSA, HPCA-TSA. A drawback of the previous formulation is that the relationships with the unsupervised feature extraction principles that underlie the Hebbian pattern discovery processes (i.e., clustering for SWTA and PCA for HPCA) are lost. In fact, we are treating upsampled and downsampled feature maps as we would do for an ordinary convolutional block, although the relationship between inputs and outputs is reversed: the upsampled feature map is now the output of the neurons, and the downsampled feature map is the input.

Therefore, we also investigate a second strategy for applying Hebbian learning in T-Conv layers, where the reconstruction block is appropriately redesigned, in order to address the shape mismatch issue. In this case, as shown in Fig. 3c, the downsampled feature map is treated as the target signal, while reconstruction and gating terms are computed from the upsampled feature map. We introduce a new reconstruction block $\rho^*(\mathbf{y}, \mathbf{w})$ which performs the following computations: (i) apply a different transformation to the upsampled feature map, depending on whether we are performing HPCA or SWTA; (ii) extract patches at different offsets from the resulting feature maps, with the desired size, stride, etc. (also known as *unfolding*); and (iii) perform matrix multiplication between the (vectorized) patches and the weight matrix. The transformations mentioned in step (i) are as follows: concerning neuron j, in the case of SWTA, set the j-th channel to 1 and the rest to 0; in the case of HPCA, the transformation is the identity for the first j channels and set the others to 0. Steps (ii) and (iii) are also equivalent to—and can be implemented efficiently as—an ordinary convolution. From now on, we call this *Transposed-Structure-Aware* (TSA) formulation of SWTA and HPCA to T-Conv layers as SWTA-TSA and HPCA-TSA, respectively.

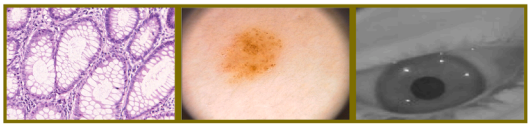
In the experimental section, we focus on SWTA-based methods, which empirically lead to better results. Indeed, for semantic segmentation tasks where pixel-wise classification is required, clustering can be a good proxy for unsupervised class discovery in the observed tensors, therefore representing a promising inductive bias for successive fine-tuning. Specifically, we consider the setting involving SWTA-TSA, which resulted in the best-performing solution. However, we also report an ablation study over the different learning models (SWTA-S, HPCA-S, SWTA-TSA, HPCA-TSA) in Section 5.4.

5. Experimental evaluation

5.1. Datasets and evaluation metrics

We assess our proposed approach on three public datasets widely used in the 2D biomedical image segmentation literature (Basak et al., 2022; Karimi et al., 2023; Valanarasu et al., 2021, 2022) featuring different imaging modalities and segmentation tasks (see Table 1 and the follow-

Table 1
Summary of datasets. We report some statistics, an image sample, and, in the last row, the associated targets exploited during the supervised training step.

	GlaS [25]	PH2 [26]	HMEPS [27]
Images	165	200	11,897
Size (px)	775×522	768×560	128×128
Segm. Task	Colorectal Cancer	Skin Lesion	Eye Pupil
Sample			
Target			

ing text for further details). Additionally, experiments on two further public datasets are reported in Appendix D.

GlaS (Sirinukunwattana et al., 2017). The Gland Segmentation in Colon Histology Images Challenge (GlaS) dataset includes 165 (H&E) stained histological 775 × 522 images of colorectal adenocarcinoma, of which 80 for training and 85 for testing.

PH2 (Mendonça et al., 2013). This dermoscopic image database comprises 200 melanocytic lesions, including 80 common nevi, 80 atypical nevi, and 40 melanomas; they are 8-bit RGB color images with a resolution of 768 × 560 pixels.

HMEPS (Mazziotti et al., 2021). The Human and Mouse Eyes for Pupil Semantic Segmentation (HMEPS) dataset is a collection of 11,897 grayscale images of humans (4285) and mice (7612) eyes illuminated using infrared (IR, 850 nm) light sources, labeled with polygons over pupil areas.

We report four metrics widely used for biomedical image segmentation (Hatamizadeh et al., 2022; Luo et al., 2021, 2022; Milletari et al., 2016), i.e., Dice Coefficient (DC), Jaccard Index (JI), 95th percentile Hausdorff Distance (95HD), and Average Surface Distance (ASD). The first two evaluators emphasize pixel-wise accuracy, while 95HD and ASD focus on boundary accuracy. DC is considered the gold standard evaluator for this task.

5.2. Comparison with SOTA

We quantitatively compare our approach against several SOTA single-stage semi-supervised techniques comprising pseudo-labeling and consistency training strategies—ADVENT (Vu et al., 2019), CCT (Ouali et al., 2020), UAMT (Yu et al., 2019), CPS (Chen et al., 2021), and URPC (Luo et al., 2022). Since these SOTA methodologies have different experimental settings, we reimplement them to run the experimental evaluation under fair conditions, using the same UNet model as the underlying architecture. Additionally, we designed two-stage pipelines as competitive baselines, featuring unsupervised pre-training based on VAEs (Kingma & Welling, 2014) and self-supervised learning (SSL) techniques (Ouyang et al., 2020; Perez-Garcia et al., 2025). These are followed by fine-tuning a semantic segmentation branch on top of the features learned during the unsupervised stage, using labeled samples. Both training stages were optimized using standard SGD or Adam optimizers. For SSL-based approaches, we specifically considered (Ouyang et al., 2020) (referred to as SuperpixSSL) and RAD-DINO (Perez-Garcia et al., 2025). Notably, the latter is the only method not relying on the UNet architecture, as it employs a transformer-based backbone (Dosovitskiy et al., 2020; Oquab et al., 2023). Finally, we include a comparison with a bio-inspired model for semantic segmentation based on spiking neural networks (SNNs), which employs a surrogate gradient method for optimization (Kim et al., 2022). Further implementation details are provided in Appendix C.

Table 2
Comparisons with SOTA on the GlaS dataset (Sirinukunwattana et al., 2017). **Bold** and *italic* indicate the best and second-best performance.

Labeled %	Method	DC (%) ↑	JI (%) ↑	95HD ↓	ASD ↓
100%	Fully Sup.	90.62 ± 0.20	82.85 ± 0.34	8.96 ± 0.34	1.76 ± 0.05
1%	VAE	68.77 ± 0.51	52.42 ± 0.59	38.33 ± 5.68	11.90 ± 1.99
	SuperpixSSL	68.51 ± 0.50	52.11 ± 0.59	39.72 ± 5.59	12.48 ± 1.88
	RAD-DINO	68.06 ± 0.17	51.59 ± 0.20	51.95 ± 8.16	15.04 ± 2.44
	SpikingNN	69.32 ± 1.01	53.07 ± 1.21	38.40 ± 5.19	11.68 ± 1.86
	ADVENT	68.92 ± 0.77	52.60 ± 0.90	40.12 ± 3.48	10.92 ± 1.04
	CCT	68.97 ± 0.73	52.65 ± 0.86	40.72 ± 3.68	11.00 ± 1.21
	UAMT	69.12 ± 0.86	52.83 ± 1.02	*37.56 ± 4.76*	*10.24 ± 1.43*
	CPS	*69.32 ± 0.59*	*53.05 ± 0.69*	38.29 ± 4.85	10.46 ± 1.37
	URPC	68.38 ± 0.44	51.96 ± 0.51	44.13 ± 2.92	12.04 ± 0.84
	Ours	**69.95 ± 1.09**	**53.82 ± 1.31**	**32.87 ± 3.36**	**9.05 ± 1.09**
2%	VAE	70.10 ± 0.95	53.99 ± 1.13	29.92 ± 4.05	8.88 ± 1.30
	SuperpixSSL	70.10 ± 1.25	54.00 ± 1.51	33.97 ± 5.02	10.24 ± 1.75
	RAD-DINO	68.52 ± 0.22	52.11 ± 0.25	48.38 ± 3.90	12.19 ± 1.51
	SpikingNN	69.90 ± 1.32	53.77 ± 1.58	36.43 ± 6.40	11.00 ± 2.31
	ADVENT	70.23 ± 1.34	54.16 ± 1.60	36.84 ± 4.92	9.65 ± 1.45
	CCT	70.05 ± 0.94	53.92 ± 1.11	31.26 ± 5.74	8.08 ± 1.71
	UAMT	69.71 ± 1.27	53.55 ± 1.52	35.35 ± 5.70	9.35 ± 1.72
	CPS	*70.60 ± 1.13*	*54.59 ± 1.35*	**29.81 ± 4.82**	**7.57 ± 1.48**
	URPC	68.67 ± 0.98	52.31 ± 1.17	42.20 ± 3.21	11.51 ± 0.95
	Ours	**71.18 ± 1.28**	**55.30 ± 1.54**	*30.07 ± 5.09*	*7.64 ± 1.56*
5%	VAE	76.16 ± 1.21	61.54 ± 1.56	23.45 ± 3.15	6.33 ± 1.06
	SuperpixSSL	75.18 ± 1.27	60.28 ± 1.63	23.19 ± 2.55	6.21 ± 0.73
	RAD-DINO	73.40 ± 0.42	57.15 ± 0.49	38.16 ± 2.64	9.72 ± 1.14
	SpikingNN	72.54 ± 1.48	56.98 ± 1.83	25.01 ± 4.52	7.22 ± 1.59
	ADVENT	75.34 ± 0.92	60.45 ± 1.17	24.74 ± 2.21	5.86 ± 0.67
	CCT	*76.33 ± 1.10*	*61.76 ± 1.43*	23.62 ± 2.67	5.54 ± 0.72
	UAMT	75.14 ± 0.65	60.19 ± 0.83	25.98 ± 1.79	6.18 ± 0.50
	CPS	76.17 ± 0.98	61.54 ± 1.28	*22.92 ± 1.84*	*5.28 ± 0.37*
	URPC	74.32 ± 1.06	59.17 ± 1.34	26.72 ± 2.79	6.59 ± 0.76
	Ours	**77.18 ± 1.05**	**62.87 ± 1.40**	**21.01 ± 2.47**	**4.86 ± 0.60**
10%	VAE	79.28 ± 1.39	65.73 ± 1.88	17.98 ± 1.35	4.66 ± 0.42
	SuperpixSSL	77.82 ± 1.67	63.78 ± 2.19	21.05 ± 3.43	5.67 ± 1.24
	RAD-DINO	76.51 ± 0.54	61.46 ± 0.65	34.74 ± 3.06	8.07 ± 1.06
	SpikingNN	77.62 ± 1.02	63.46 ± 1.36	19.65 ± 1.04	5.02 ± 0.32
	ADVENT	78.08 ± 0.82	64.06 ± 1.10	21.76 ± 1.81	4.87 ± 0.44
	CCT	*80.36 ± 1.05*	*67.19 ± 1.44*	*17.80 ± 1.05*	*3.91 ± 0.36*
	UAMT	79.31 ± 0.77	65.72 ± 1.04	18.37 ± 1.38	4.12 ± 0.25
	CPS	80.35 ± 1.11	67.16 ± 1.56	18.73 ± 1.58	4.23 ± 0.46
	URPC	78.59 ± 1.39	64.78 ± 1.85	21.57 ± 3.05	5.03 ± 0.93
	Ours	**80.77 ± 0.77**	**67.77 ± 1.08**	**17.64 ± 0.82**	**3.87 ± 0.21**
20%	VAE	83.22 ± 1.06	71.30 ± 1.55	15.20 ± 0.82	3.61 ± 0.24
	SuperpixSSL	81.33 ± 1.36	68.60 ± 1.91	17.13 ± 1.54	4.16 ± 0.37
	RAD-DINO	82.18 ± 0.46	68.47 ± 0.56	28.12 ± 1.35	6.9 ± 0.53
	SpikingNN	81.60 ± 0.48	68.92 ± 0.69	16.22 ± 0.59	4.03 ± 0.18
	ADVENT	81.20 ± 0.80	68.38 ± 1.13	15.96 ± 1.11	3.55 ± 0.24
	CCT	*84.22 ± 0.84*	*72.76 ± 1.25*	*14.26 ± 0.79*	*2.98 ± 0.13*
	UAMT	83.03 ± 0.69	71.00 ± 1.00	14.56 ± 0.68	3.22 ± 0.19
	CPS	83.90 ± 0.51	72.27 ± 0.77	14.29 ± 0.32	3.09 ± 0.11
	URPC	82.34 ± 2.07	70.12 ± 2.84	16.74 ± 2.16	3.64 ± 0.57
	Ours	**84.50 ± 0.50**	**73.17 ± 0.75**	**13.96 ± 0.55**	**2.93 ± 0.11**

We performed experiments considering several semi-supervised setups, i.e., we supervised the models with 1%, 2%, 5%, 10%, and 20% labeled images. We report results showing the mean over ten independent runs together with 90% confidence intervals. Qualitative results for our proposed method can be found in Fig. 5, while additional qualitative examples for competing techniques are provided in Appendix E.

GlaS. The results on the GlaS dataset are given in Table 2. Our approach outperforms previous works, considering all the metrics in almost all the considered settings. Concerning DC, we outperform SOTA techniques by about 1%-2%, depending on the considered data regime.

PH2. Table 3 illustrates the results on the PH2 dataset. Even in this case, our approach achieves the best results in almost all the settings, except for 95HD and ASD in some training data regimes where, anyway, we obtain the second highest results. Concerning DC, we outper-

Table 3
Comparisons with SOTA on the PH2 dataset (Mendonça et al., 2013). **Bold** and *italic* indicate the best and second-best performance.

Labeled %	Method	DC (%) ↑	JI (%) ↑	95HD ↓	ASD ↓
100%	Fully Sup.	92.44 ± 0.38	85.96 ± 0.66	6.77 ± 0.72	2.34 ± 0.12
1%	VAE	75.83 ± 0.96	61.09 ± 1.22	32.37 ± 6.10	10.44 ± 2.05
	SuperpixSSL	70.78 ± 2.12	54.87 ± 2.60	33.79 ± 9.46	12.66 ± 3.01
	RAD-DINO	*77.18 ± 3.53*	*63.19 ± 7.18*	34.16 ± 6.31	12.50 ± 3.29
	SpikingNN	69.57 ± 2.82	53.55 ± 3.39	27.45 ± 5.63	11.22 ± 1.93
	ADVENT	73.24 ± 2.32	57.92 ± 2.87	25.96 ± 7.74	10.01 ± 3.03
	CCT	73.42 ± 1.58	58.06 ± 1.98	27.40 ± 7.09	10.27 ± 2.53
	UAMT	74.72 ± 1.45	59.70 ± 1.84	25.06 ± 6.78	*8.20 ± 1.31*
	CPS	76.07 ± 2.12	61.50 ± 2.71	28.79 ± 7.93	10.03 ± 2.81
	URPC	71.23 ± 1.95	55.39 ± 2.33	**23.17 ± 8.10**	10.41 ± 2.59
	Ours	**78.16 ± 3.04**	**64.25 ± 3.69**	*24.19 ± 8.57*	**8.18 ± 2.39**
2%	VAE	78.78 ± 1.61	65.06 ± 2.18	22.29 ± 4.73	7.24 ± 1.06
	SuperpixSSL	77.61 ± 2.47	63.59 ± 3.28	19.13 ± 1.88	7.16 ± 0.72
	RAD-DINO	79.66 ± 2.65	66.22 ± 3.69	32.78 ± 5.21	10.45 ± 2.50
	SpikingNN	72.53 ± 3.88	57.32 ± 4.67	25.63 ± 6.47	10.09 ± 2.45
	ADVENT	79.34 ± 2.63	65.95 ± 3.55	17.69 ± 3.41	6.62 ± 0.96
	CCT	*80.13 ± 1.41*	*66.90 ± 1.96*	**13.73 ± 0.88**	**5.82 ± 0.47**
	UAMT	79.76 ± 1.26	66.38 ± 1.74	15.05 ± 1.89	6.10 ± 0.59
	CPS	79.64 ± 2.72	66.39 ± 3.71	14.97 ± 1.92	6.33 ± 0.88
	URPC	78.88 ± 1.69	65.22 ± 2.33	*14.68 ± 1.33*	6.90 ± 0.76
	Ours	**80.90 ± 1.66**	**67.97 ± 2.36**	17.33 ± 2.30	*6.08 ± 0.38*
5%	VAE	82.00 ± 0.94	69.53 ± 1.33	18.33 ± 2.23	6.02 ± 0.54
	SuperpixSSL	81.78 ± 1.38	69.23 ± 1.95	15.41 ± 2.18	5.74 ± 0.57
	RAD-DINO	*85.05 ± 2.43*	*74.05 ± 3.62*	17.03 ± 4.07	6.9 ± 1.3
	SpikingNN	80.29 ± 1.79	67.18 ± 2.43	18.78 ± 3.58	6.85 ± 0.74
	ADVENT	81.89 ± 1.02	69.37 ± 1.44	13.72 ± 1.85	5.45 ± 0.34
	CCT	82.78 ± 1.15	70.66 ± 1.63	13.64 ± 1.71	5.27 ± 0.35
	UAMT	83.75 ± 1.11	72.08 ± 1.64	*12.29 ± 0.77*	*4.81 ± 0.26*
	CPS	82.86 ± 1.18	70.78 ± 1.73	15.00 ± 3.41	5.53 ± 0.68
	URPC	83.41 ± 2.17	71.65 ± 3.14	**10.78 ± 0.99**	4.86 ± 0.43
	Ours	**85.77 ± 1.51**	**75.13 ± 2.28**	14.10 ± 0.87	**4.78 ± 0.30**
10%	VAE	84.29 ± 1.44	72.91 ± 2.15	15.06 ± 2.18	5.22 ± 0.61
	SuperpixSSL	85.21 ± 1.15	74.28 ± 1.77	12.27 ± 1.42	4.53 ± 0.28
	RAD-DINO	88.01 ± 0.81	78.02 ± 1.31	16.33 ± 0.93	5.45 ± 0.42
	SpikingNN	83.86 ± 1.96	72.36 ± 2.90	12.65 ± 1.19	4.90 ± 0.54
	ADVENT	84.94 ± 0.90	73.85 ± 1.35	*11.21 ± 1.41*	4.48 ± 0.41
	CCT	87.93 ± 1.96	78.46 ± 3.14	13.05 ± 3.50	3.88 ± 1.49
	UAMT	87.37 ± 0.58	77.59 ± 0.92	12.17 ± 1.64	4.43 ± 0.25
	CPS	84.33 ± 0.82	72.93 ± 1.21	13.19 ± 1.71	4.85 ± 0.37
	URPC	*88.06 ± 0.40*	*78.89 ± 0.45*	**8.95 ± 1.89**	**3.71 ± 0.36**
	Ours	**88.26 ± 0.51**	**79.00 ± 0.82**	13.04 ± 2.39	*4.35 ± 0.63*
20%	VAE	87.93 ± 1.19	78.52 ± 1.90	12.13 ± 2.11	4.12 ± 0.58
	SuperpixSSL	88.31 ± 0.94	79.10 ± 1.52	10.39 ± 1.05	3.75 ± 0.27
	RAD-DINO	91.52 ± 0.61	84.38 ± 1.02	14.69 ± 1.42	5.08 ± 0.44
	SpikingNN	89.32 ± 0.84	80.73 ± 1.39	9.12 ± 0.77	3.29 ± 0.20
	ADVENT	86.30 ± 0.87	75.92 ± 1.33	10.33 ± 1.36	3.97 ± 0.33
	CCT	89.95 ± 0.57	81.74 ± 0.94	8.21 ± 0.60	3.04 ± 0.10
	UAMT	88.95 ± 0.64	80.12 ± 1.04	10.52 ± 1.58	3.56 ± 0.28
	CPS	86.49 ± 0.97	76.23 ± 1.52	10.56 ± 1.15	4.14 ± 0.37
	URPC	*91.73 ± 0.80*	*84.71 ± 1.36*	**6.35 ± 0.13**	*2.48 ± 0.11*
	Ours	**91.99 ± 0.82**	**84.92 ± 1.38**	*8.01 ± 2.13*	**2.13 ± 0.87**

Table 4
Comparisons with SOTA on the HMEPS dataset (Mazziotti et al., 2021). **Bold** and *italic* indicate the best and second-best performance.

Labeled %	Method	DC (%) ↑	JI (%) ↑	95HD ↓	ASD ↓
100%	Fully Sup.	96.98 ± 0.42	94.70 ± 0.43	0.06 ± 0.05	0.03 ± 0.00
1%	VAE	89.53 ± 1.94	82.39 ± 3.30	7.14 ± 5.69	2.07 ± 2.36
	SuperpixSSL	87.45 ± 3.13	77.80 ± 4.94	18.67 ± 9.46	3.91 ± 3.20
	RAD-DINO	86.69 ± 0.83	75.06 ± 1.19	7.27 ± 0.26	3.40 ± 0.16
	SpikingNN	80.98 ± 2.63	65.20 ± 3.11	7.54 ± 1.05	3.30 ± 0.44
	ADVENT	90.24 ± 2.74	82.25 ± 4.52	3.95 ± 2.29	1.24 ± 0.72
	CCT	**91.09 ± 2.50**	**84.03 ± 4.24**	**2.63 ± 0.51**	**0.85 ± 0.46**
	UAMT	90.18 ± 0.77	82.12 ± 1.27	4.19 ± 2.20	1.15 ± 0.39
	CPS	90.39 ± 0.55	82.49 ± 0.91	4.40 ± 0.78	1.16 ± 0.11
	URPC	89.15 ± 0.79	80.45 ± 1.28	5.96 ± 1.92	1.46 ± 0.32
	Ours	*90.75 ± 0.50*	*83.07 ± 2.19*	*3.70 ± 1.45*	*1.07 ± 0.51*
2%	VAE	91.51 ± 2.42	84.59 ± 3.77	3.82 ± 2.44	1.10 ± 0.64
	SuperpixSSL	88.47 ± 4.53	79.53 ± 7.16	21.04 ± 10.75	5.26 ± 3.34
	RAD-DINO	85.81 ± 0.52	75.16 ± 0.80	5.98 ± 0.36	2.63 ± 0.12
	SpikingNN	83.83 ± 2.61	70.24 ± 2.45	6.98 ± 2.23	3.05 ± 1.81
	ADVENT	91.35 ± 1.93	84.14 ± 3.19	2.72 ± 0.61	0.82 ± 0.23
	CCT	91.48 ± 2.45	84.34 ± 4.19	2.48 ± 1.53	0.83 ± 0.41
	UAMT	*92.42 ± 0.09*	*86.06 ± 0.16*	*2.35 ± 0.90*	*0.72 ± 0.13*
	CPS	91.07 ± 0.28	83.60 ± 0.48	3.11 ± 0.81	0.93 ± 0.11
	URPC	89.78 ± 0.64	81.47 ± 1.05	3.99 ± 0.63	1.14 ± 0.11
	Ours	**92.60 ± 1.20**	**86.21 ± 2.09**	**2.25 ± 0.69**	**0.70 ± 0.11**
5%	VAE	93.09 ± 0.56	87.09 ± 0.98	2.10 ± 0.27	0.65 ± 0.07
	SuperpixSSL	91.34 ± 2.35	84.16 ± 3.96	4.36 ± 3.10	1.25 ± 0.71
	RAD-DINO	88.72 ± 0.39	79.73 ± 0.63	4.87 ± 0.18	2.09 ± 0.07
	SpikingNN	87.96 ± 0.84	77.40 ± 1.29	3.05 ± 0.29	1.26 ± 0.12
	ADVENT	92.31 ± 1.07	85.78 ± 1.84	2.38 ± 0.72	0.79 ± 0.17
	CCT	*93.25 ± 0.92*	*87.38 ± 1.57*	*1.59 ± 0.36*	*0.59 ± 0.11*
	UAMT	93.03 ± 1.18	87.02 ± 2.00	2.08 ± 0.93	0.65 ± 0.16
	CPS	92.89 ± 0.41	86.72 ± 0.72	2.25 ± 0.79	0.69 ± 0.14
	URPC	90.85 ± 0.72	83.23 ± 1.22	4.11 ± 2.84	1.05 ± 0.39
	Ours	**93.51 ± 0.25**	**87.81 ± 0.45**	**1.35 ± 0.31**	**0.47 ± 0.03**
10%	VAE	93.38 ± 0.37	87.58 ± 0.65	1.72 ± 0.34	0.59 ± 0.09
	SuperpixSSL	92.82 ± 2.01	86.65 ± 3.44	2.94 ± 2.51	0.84 ± 0.62
	RAD-DINO	90.10 ± 0.42	81.99 ± 0.70	4.20 ± 0.21	1.79 ± 0.10
	SpikingNN	90.31 ± 0.41	81.69 ± 0.68	2.38 ± 0.15	1.06 ± 0.06
	ADVENT	92.56 ± 0.96	86.20 ± 1.65	2.42 ± 0.85	0.75 ± 0.19
	CCT	*93.45 ± 0.05*	*87.70 ± 0.09*	*1.55 ± 0.79*	*0.57 ± 0.18*
	UAMT	93.14 ± 0.47	87.16 ± 0.82	1.67 ± 0.43	0.58 ± 0.08
	CPS	92.83 ± 0.46	86.63 ± 0.81	1.83 ± 0.37	0.62 ± 0.08
	URPC	90.96 ± 0.83	83.44 ± 1.40	2.31 ± 0.58	0.83 ± 0.13
	Ours	**93.68 ± 0.28**	**88.12 ± 0.50**	**1.37 ± 0.19**	**0.49 ± 0.04**
20%	VAE	93.42 ± 0.19	87.66 ± 0.34	1.78 ± 0.23	0.58 ± 0.04
	SuperpixSSL	93.18 ± 0.24	87.23 ± 0.42	1.76 ± 0.24	0.60 ± 0.03
	RAD-DINO	91.17 ± 0.23	83.77 ± 0.38	3.66 ± 0.16	1.59 ± 0.07
	SpikingNN	91.80 ± 0.30	84.85 ± 0.51	2.02 ± 0.11	0.91 ± 0.04
	ADVENT	93.04 ± 0.59	87.00 ± 1.02	1.73 ± 0.37	0.60 ± 0.09
	CCT	93.30 ± 0.20	87.45 ± 0.36	1.51 ± 0.28	*0.54 ± 0.05*
	UAMT	*93.42 ± 0.82*	*87.68 ± 1.55*	1.54 ± 0.46	0.56 ± 0.12
	CPS	93.04 ± 0.12	86.99 ± 0.22	*1.48 ± 0.16*	0.55 ± 0.02
	URPC	91.30 ± 1.44	84.05 ± 2.47	2.79 ± 1.28	0.88 ± 0.20
	Ours	**93.82 ± 0.16**	**88.35 ± 0.29**	**1.26 ± 0.13**	**0.46 ± 0.01**

form SOTA techniques up to 3%, depending on the considered data regime.

HMEPS. The outcomes on the HMEPS dataset are presented in Table 4. Again, our approach obtains improved SOTA performance in almost all the considered settings. However, we did not reach the best results with regime 1%. We deem that this behavior can be linked to the intrinsic characteristics of this specific scenario, i.e., the HMEPS dataset provides a bigger set of labeled samples, even for the considered low training data regimes, making the unsupervised step less significant.

Discussion. While the proposed Hebbian learning framework demonstrates strong sample efficiency across multiple biomedical datasets, it is not without limitations. The approach inherently relies on the unsupervised discovery of salient low-level features from the data's intrinsic structure. As a result, its effectiveness depends on the complexity and homogeneity of the visual features present in the unlabeled corpus. This limitation helps explain its relative underperformance in certain scenarios—for example, the extremely low-label (1%) setting on HMEPS. Some tasks, such as segmenting a singular, well-defined, and highly contrasted pupil structure, are inherently less complex than others, like segmenting irregular skin lesions or intricate cellular boundaries. In cases where the dataset size is already substantial (e.g., 2000 images in HMEPS), even a 1% subset may provide a sufficiently large and representative set of labeled examples for a relatively simple task. In such situations, conventional supervised methods can quickly learn the task via backpropagation. By contrast, the unsupervised Hebbian pretraining—designed to build generic feature hierarchies from large unlabeled data—offers diminished marginal utility when the task is characterized by high contrast, consistent shape, and minimal background clutter. In fact, it may even slightly hinder performance if it biases the model toward priors that are suboptimal for such simple feature distributions. This observation, empirically supported by our results, suggests that the primary advantage of our method emerges in

Table 5
Ablation on the temperature hyperparameter. Mean ± 90% CI are reported. The best results are in **bold**.

Dataset (20% Labeled)	Temperature	DC (%) ↑
GlaS	1	82.34 ± 1.06
	5	82.90 ± 0.68
	10	83.01 ± 0.68
	20	83.39 ± 0.58
	50	83.84 ± 0.71
	75	84.15 ± 0.50
	100	**84.50 ± 0.50**
PH2	1	86.57 ± 1.51
	5	86.81 ± 1.16
	10	89.00 ± 0.98
	20	**91.99 ± 0.82**
	50	88.48 ± 0.72
	75	89.05 ± 1.06
	100	89.15 ± 0.86
HMEPS	1	92.10 ± 0.38
	5	92.84 ± 0.85
	10	93.10 ± 0.94
	20	**93.82 ± 0.16**
	50	93.41 ± 0.23
	75	93.38 ± 0.34
	100	92.98 ± 0.85

Table 6
Ablation on the type of Hebbian learning algorithm considering the GlaS dataset. The best results are in **bold**.

Labeled%	Method	DC (%) ↑
1%	HPCA-S	66.18 ± 1.60
	HPCA-TSA	65.18 ± 3.63
	SWTA-S	68.26 ± 1.67
	SWTA-TSA	**69.95 ± 1.09**
2%	HPCA-S	68.38 ± 1.77
	HPCA-TSA	69.53 ± 1.27
	SWTA-S	69.79 ± 1.20
	SWTA-TSA	**71.18 ± 1.28**
5%	HPCA-S	75.19 ± 1.63
	HPCA-TSA	73.62 ± 2.58
	SWTA-S	74.89 ± 1.54
	SWTA-TSA	**77.18 ± 1.05**
10%	HPCA-S	80.02 ± 1.23
	HPCA-TSA	79.73 ± 1.12
	SWTA-S	79.82 ± 1.06
	SWTA-TSA	**80.77 ± 0.77**
20%	HPCA-S	83.20 ± 0.58
	HPCA-TSA	83.61 ± 0.75
	SWTA-S	84.00 ± 0.62
	SWTA-TSA	**84.50 ± 0.50**

complex segmentation tasks, where labeled data is truly scarce and the visual features are heterogeneous and high-dimensional.

5.3. Hebbian initialization of single-stage methods

We report the results obtained by using our unsupervised SWTA-TSA pre-training as initialization for the considered single-stage pseudo-labeling and consistency-based semi-supervised methods. Results are shown in Fig. 4. For better readability, we illustrate only the performance in terms of the gold standard metric, i.e., the DC. We can observe that, in most cases, the proposed initialization helps achieve significant improvements. However, we can still notice a small worsening in some cases. We deem that this can be expected because a given initialization, although effective for some methods and specific data distributions/data regimes, can be sub-optimal for other techniques, which instead require starting with small random weights to guarantee the stability and convergence of the training process.

5.4. Ablation studies

Softmax temperature hyperparameter. We perform an ablation study varying the temperature hyperparameter defined in Eq. (2). We show results concerning SWTA-TSA over the three datasets with the regime at 20% in Table 5. We observe that, concerning the GlaS dataset, increasing temperature from 1 to 100 has a positive impact on performance until a plateau is reached; instead, we obtained the best performance with a temperature of 20 for the PH2 and HMEPS datasets. Furthermore, we note that, while some temperature settings yield statistically significant improvements (e.g., PH2), in other cases confidence intervals overlap. Nonetheless, selecting the configuration with the best average performance remains a valid strategy, given the smoothness of the hyperparameter landscape.

Hebbian variants. Table 6 concisely ablates the results obtained by exploiting the different Hebbian learning strategies introduced in Section 4.2. Specifically, we report the outcomes in terms of the Dice Coefficient for the GlaS dataset (the most popular and challenging among those used) considering different percentages of label availability. In all cases, our SWTA-TSA Hebbian unsupervised learning formulation achieves the best performance, motivating its selection as our preferred choice.

Hebbian unsupervised first stage. We conduct an ablation study to assess the Hebbian unsupervised pre-training independently, focusing on both DC and training time. Specifically, for DC evaluation, we follow established literature and employ linear probing, where a linear classifier is trained on top of the frozen learned features (Gansbeke et al., 2021; Ji et al., 2019). Table 7 presents the DC results on the GlaS dataset, comparing our approach with the other unsupervised pre-training competitors, i.e., VAE (Kingma & Welling, 2014) and SuperpixSSL (Ouyang et al., 2020). We exclude RAD-DINO (Perez-Garcia et al., 2025) from this comparison due to its unfair advantage, as its backbone was trained on a substantially larger dataset (see Appendix C for details). Our approach achieves the best performance, demonstrating its superiority. In terms of training time, we observe that our method offers competitive efficiency compared to the backpropagation-based VAE, while achieving significantly better segmentation metrics-both under linear probing and during second-stage semi-supervised fine-tuning. By contrast, SuperpixSSL exhibits considerably lower training efficiency, primarily due to the computational overhead associated with proxy target generation.

5.5. Experiments with volumetric images

To prove that our approach can be easily extended to 3D medical images, we report some outcomes using the Left Atrial (LA) dataset (Xiong et al., 2021). Specifically, it contains 100 3D MRI images from the 2018 atrial segmentation challenge; following the literature, we use 80 images for training and 20 for testing.

The results of our evaluation are shown in Table 8. Our approach ranks best or second best most of the time, and our Hebbian initialization yields benefits to one-stage SOTA methods. Qualitative results can instead be found in Fig. 6

6. Conclusion and future works

In this work, we presented a novel two-stage semi-supervised approach for semantic segmentation, leveraging bio-inspired learning models for a first unsupervised pre-training step, followed by a second supervised fine-tuning phase. Our core contribution is the formulation of Hebbian learning rules for transpose-convolutional layers, constituting the upsampling path of many popular semantic segmentation architectures. Experiments over several biomedical image segmentation benchmarks

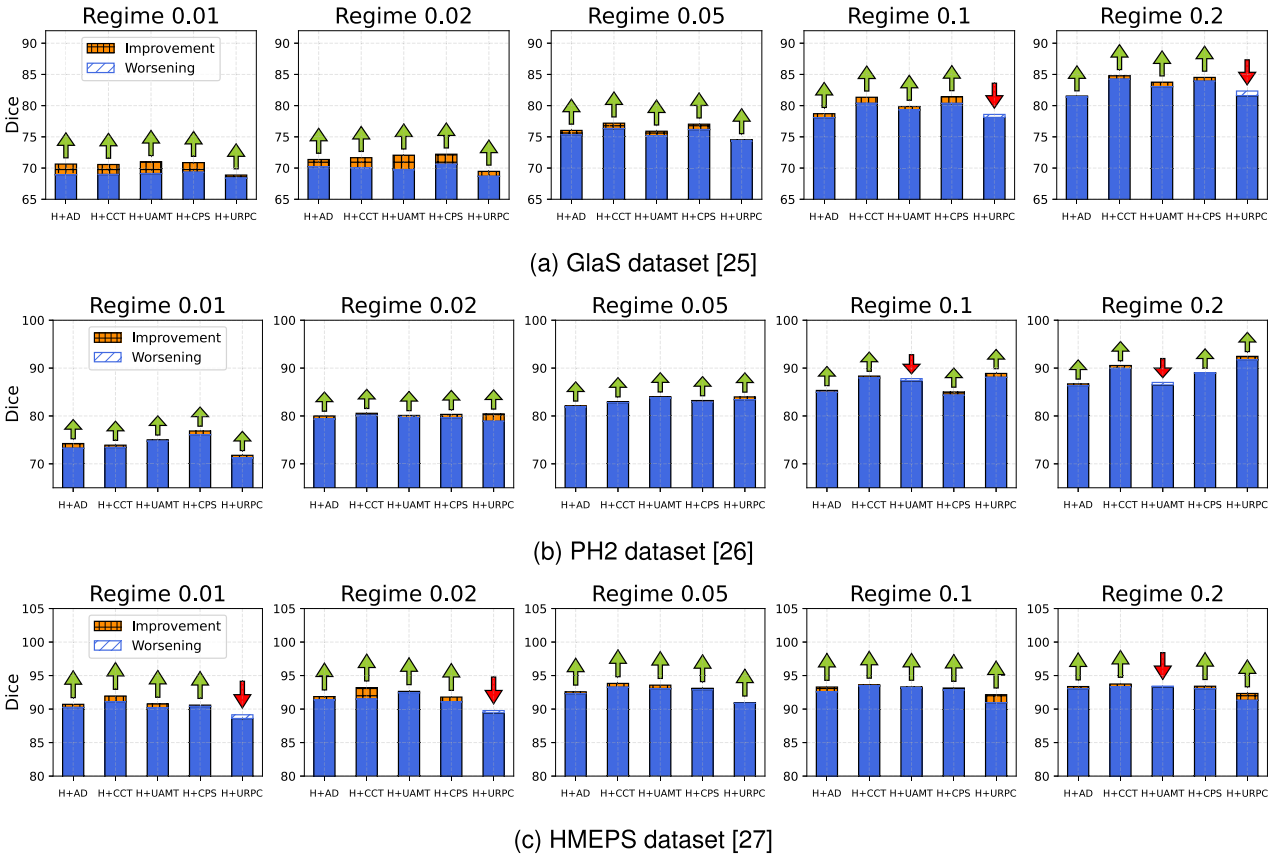


Fig. 4. Performance changes (with green and red arrows) obtained with single-stage SOTA semi-supervised approaches initialized with our unsupervised Hebbian pre-training compared to initialization from scratch (blue bar). Each row corresponds to a dataset, while each column to a different degree of label availability. We report DC values embedded in the most convenient range for best readability.

Table 7
Ablation of the first stage of our semi-supervised pipeline on the GlaS dataset. The best results are in **bold**.

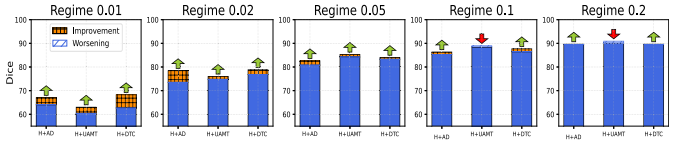
Method	DC (%) ↑	Training Time ↓
VAE	57.5	6m 6s
SuperpixSSL	44.9	916m 45s
Ours	**59.4**	**6m 1s**

using different degrees of labeled data demonstrated the effectiveness of our methodology compared to other SOTA approaches. Furthermore, we also explored combinations of our Hebbian pre-training approaches with existing pseudo-labeling and consistency-based semi-supervised methods, resulting in performance improvements. These findings hold significant practical relevance, especially considering the extreme cost required for collecting annotated data, particularly in the biomedical domain, and because bio-inspired algorithms can be advantageous for developing biologically plausible models.

The proposed methodology still presents some areas for improvement, which can be addressed in future works. One such future work is the extension of the proposed Hebbian learning framework to additional conventional architectural components, including transformer-based models, in order to further improve scalability and generalization across neural network paradigms. Another compelling direction for future research would be to extend the two-stage pipeline proposed here

Table 8
Experiments with 3D images on the LA dataset (Xiong et al., 2021).

Labeled %	Method	DC (%) ↑	JI (%) ↑	95HD ↓	ASD ↓
100%	Fully Sup.	91.76 ± 0.11	84.77 ± 0.19	5.75 ± 0.84	1.69 ± 0.06
1%	ADVENT	64.00 ± 3.55	47.09 ± 3.84	37.15 ± 4.89	9.55 ± 0.69
	UAMT	60.42 ± 4.56	43.39 ± 4.56	40.92 ± 5.49	10.37 ± 1.65
	DTC	*62.63 ± 4.77*	*45.66 ± 4.99*	**35.47 ± 3.34**	**9.41 ± 0.61**
	Ours	**65.08 ± 4.74**	**48.26 ± 4.54**	*40.82 ± 3.66*	*10.15 ± 1.41*
2%	ADVENT	73.53 ± 4.09	58.36 ± 6.96	31.35 ± 6.85	7.50 ± 1.72
	UAMT	74.80 ± 5.18	59.93 ± 6.42	28.05 ± 8.61	6.52 ± 2.18
	DTC	**76.87 ± 4.50**	**62.51 ± 5.88**	*25.74 ± 2.94*	*5.82 ± 0.74*
	Ours	*76.42 ± 1.87*	*61.86 ± 2.46*	**24.66 ± 2.80**	**5.73 ± 0.43**
5%	ADVENT	80.80 ± 3.73	67.84 ± 5.33	23.12 ± 5.78	5.40 ± 1.17
	UAMT	**84.22 ± 3.67**	**72.80 ± 5.41**	**15.97 ± 2.28**	**3.73 ± 0.42**
	DTC	83.28 ± 2.97	71.43 ± 4.33	17.55 ± 3.80	4.19 ± 0.94
	Ours	*83.42 ± 1.92*	*71.57 ± 2.82*	*17.40 ± 5.38*	*4.18 ± 0.60*
10%	ADVENT	85.33 ± 1.42	74.41 ± 2.18	17.72 ± 0.98	3.79 ± 0.44
	UAMT	**88.95 ± 0.22**	**80.09 ± 0.37**	**8.13 ± 0.02**	**2.21 ± 0.06**
	DTC	86.43 ± 1.73	76.13 ± 2.69	13.71 ± 4.95	3.22 ± 0.74
	Ours	*87.11 ± 0.69*	*77.17 ± 1.07*	*13.63 ± 2.16*	*3.21 ± 0.11*
20%	ADVENT	*89.51 ± 0.26*	*81.01 ± 0.42*	9.61 ± 1.34	2.43 ± 0.23
	UAMT	**90.91 ± 0.64**	**83.34 ± 1.07**	**6.09 ± 1.00**	**1.82 ± 0.34**
	DTC	89.46 ± 1.45	80.93 ± 2.36	*8.75 ± 2.29*	*2.25 ± 0.11*
	Ours	89.17 ± 1.25	80.45 ± 2.05	11.92 ± 7.96	2.66 ± 0.49



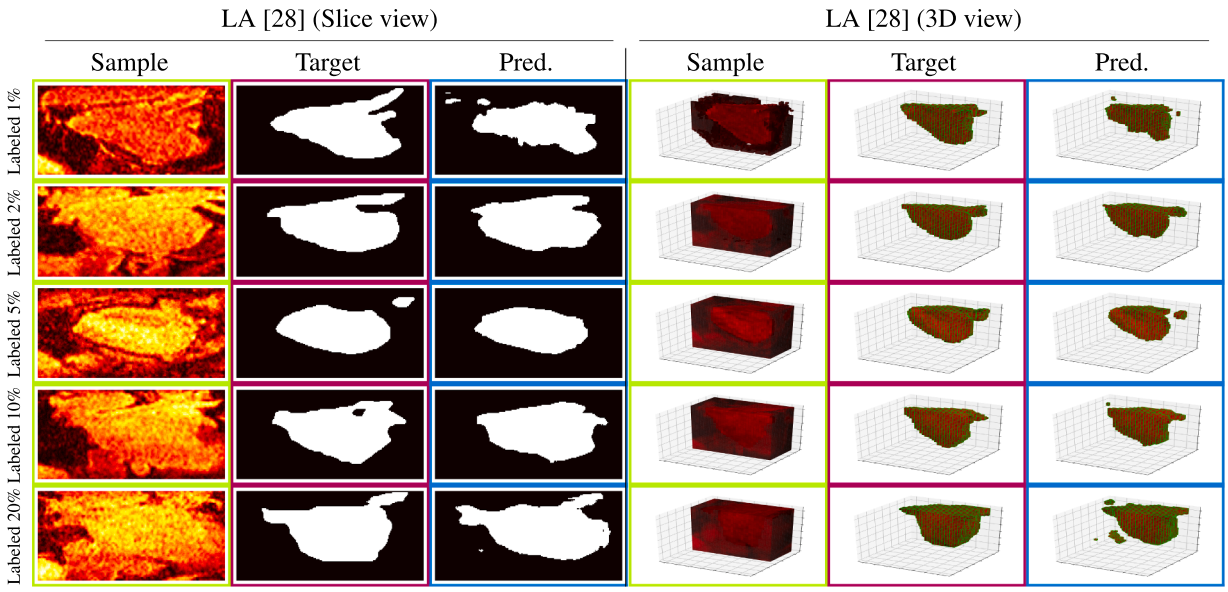


Fig. 5. Qualitative results from our semi-supervised approach based on Hebbian SWTA-TSA. Each row corresponds to a different percentage of label availability; each column corresponds to a different dataset and includes a triplet sample-target-prediction.

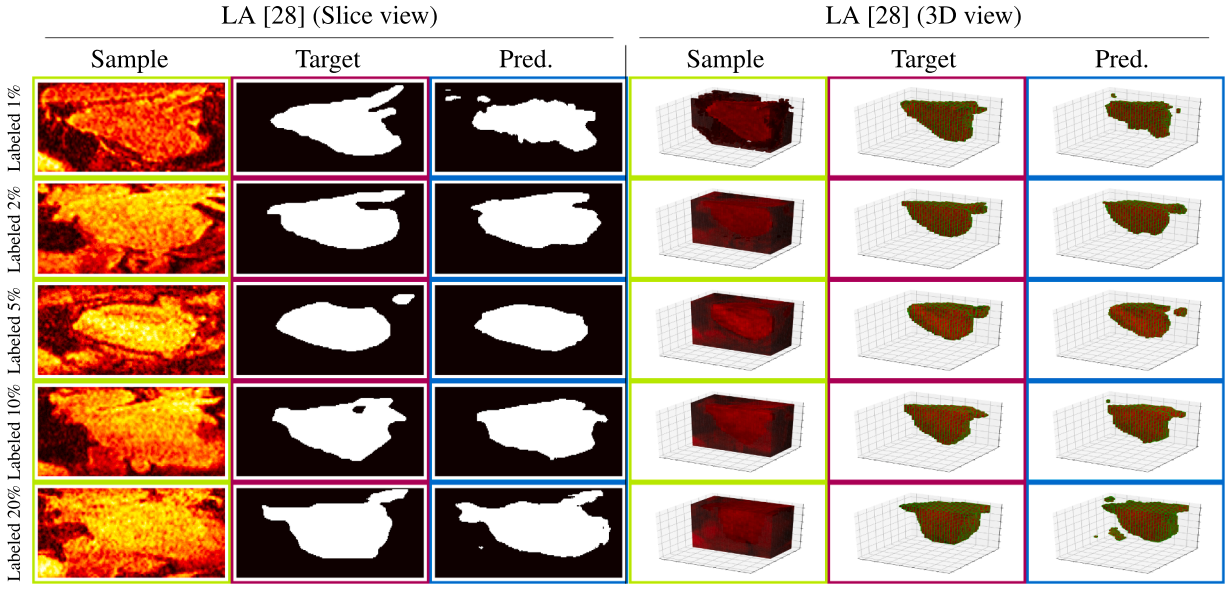


Fig. 6. Qualitative results with 3D images on the LA dataset (Xiong et al., 2021). We provide two views for better readability—a slide view and a 3D view.

to SNNs—potentially by reformulating Hebbian principles within the framework of Spike-Timing Dependent Plasticity (STDP).

CRediT authorship contribution statement

Luca Ciampi: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Gabriele Lagani:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Giuseppe Amato:** Writing – review & editing, Supervision, Project administration, Funding acquisition; **Fabrizio Falchi:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially funded by: Spoke 8, Tuscany Health Ecosystem (THE) Project (CUP B83C22003930001), funded by the National Recovery and Resilience Plan (NRRP), within the NextGeneration Europe (NGEU) Program; SUN – Social and hUman ceNtered XR (EC, Horizon Europe No. 101092612). We acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support.

Appendix A. Hebbian Learning Background

The methodology outlined in our work draws inspiration from the biological processes of synaptic adaptation and learning. While Hebbian theory has its roots in neurobiology, a background in neuroscience is not required to grasp the principles underlying these methodologies. Interestingly, mathematical models of Hebbian learning reveal surprising parallels with well-known machine learning concepts, such as clustering, Principal Component Analysis (PCA), and other mechanisms for unsupervised feature extraction. In the following sections, we provide additional context to enhance the reader's understanding of how certain learning principles emerge from the proposed learning rules. Specifically, we introduced two learning principles: Soft-Winner-Takes-All (SWTA) and Hebbian Principal Component Analysis (HPCA). We begin by examining SWTA in greater depth, highlighting its connections to standard centroid-based clustering. Subsequently, we delve into HPCA, demonstrating how this synaptic model facilitates the identification of principal components from data.

Soft-Winner-Takes-All (SWTA). The SWTA weights update rule expressed in Eq. (2) is reported again, for convenience, hereafter:

$$\Delta w_{i,j} = \eta \, \mathrm{softmax}(y_1, y_2, \ldots)_j \, (x_i - w_{i,j}), \tag{A.1}$$

For a given neuron i, we can distinguish two multiplicative components. The first component includes the softmax, together with the learning rate. It is a scalar (one term for each neuron) that essentially modulates the length of the weight update step for the given neuron. The second component is $x_i - w_{i,j}$, which is instead a vector. This is the direction of the weight update. Intuitively, the neuron takes a small update step in the direction that links its weight vector $w_{i,j}$ with the input x_i. When this process is repeated over and over for many inputs, the weight vector eventually converges to the centroid of the observed inputs (Fig. A.7). Furthermore, since the softmax operation assigns modulation coefficients close to 1 for highly active neurons and near 0 for less active ones, this mechanism enables different neurons to specialize in distinct input clusters, as neurons that take a larger step toward a specific input are more likely to produce stronger responses when similar inputs are encountered in the future. Fig. A.8 illustrates a practical example of weights and feature vectors extracted from a UNet backbone trained on the GlaS dataset—an architecture and dataset for semantic segmentation used in the experiments presented in this work. It showcases the evolution of these vectors across different training epochs, following the SWTA approach. Further details about the data processing underlying this visualization are provided in Appendix B.

Hebbian Principal Component Analysis (HPCA). For convenience, we report again Eq. (3) concerning the HPCA weights update hereafter:

$$\Delta w_{i,j} = \eta \, y_j \, (x_i - \sum_{k=1}^{j} y_k w_{i,k}). \tag{A.2}$$

The learning process underlying this rule is less intuitive to visualize compared to the SWTA case, but valuable insights can be gained by examining the conditions required for convergence to a stochastic equilibrium. Specifically, for this equilibrium to be achieved, the average weight update across all inputs must equal zero. Mathematically, this can be written as:

$$\mathbb{E}[\Delta w_{i,1}] = \mathbb{E}[\eta \, y_1 \, (x_i - y_1 w_{i,1})] = 0, \tag{A.3}$$

where the equation refers to neuron 1 in particular, but the following arguments apply also to any other neuron. By rearranging the terms, we obtain:

$$\mathbb{E}[y_1 x_i] = \mathbb{E}[y_1 y_1 w_{i,1}]. \tag{A.4}$$

At this point we can recall the input-output relationship of the neuron, i.e., $y_1 = \sum_k x_k w_{k,1}$, and substitute it in the previous equation as follows:

$$\mathbb{E}[\sum_k x_k w_{k,1} x_i] = \mathbb{E}[\sum_k x_k w_{k,1} \sum_h x_h w_{h,1} w_{i,1}]. \tag{A.5}$$

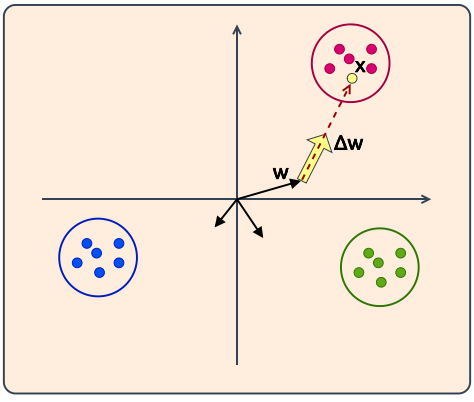
Since the sum can commute with the expectation, and since $w_{i,j}$ does not depend on x_i, after rearranging the terms we can derive the following equation:

$$\sum_k w_{k,1} \mathbb{E}[x_k x_i] = \sum_{k,h} w_{k,1} \mathbb{E}[x_k x_h] w_{h,1} w_{i,1}. \tag{A.6}$$

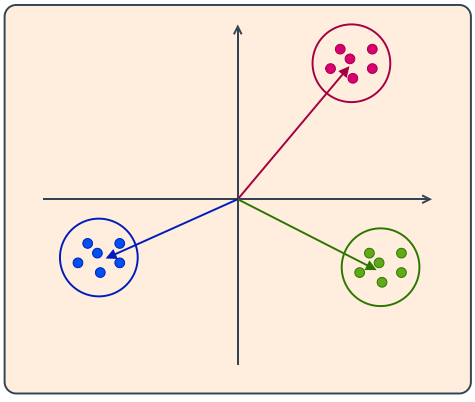
It can be noticed that $\mathbb{E}[x_i x_k]$ is the input data covariance matrix, while $\sum_{k,h} w_{k,1} \mathbb{E}[x_k x_h] w_{h,1}$ is a scalar. By calling the first term as $C_{i,k}$ and the second term as λ, we can simplify the equation to:

$$\sum_k w_{k,1} C_{i,k} = \lambda w_{i,1}. \tag{A.7}$$

This is a familiar equation of the eigenvalues and eigenvectors of the matrix $C_{i,k}$. In other words, this tells us that, according to the synaptic dynamics defined in Eq. (A.2), equilibrium of neuron 1 is achieved

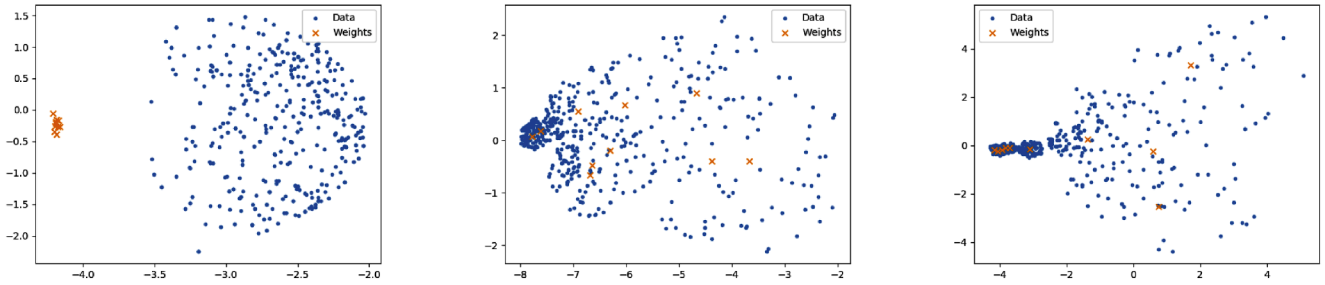


(a) Starting from a random initialization of weight vectors, inputs are presented to the neurons. The weight vectors take an update step towards the input position in the data plane.



(b) After multiple iterations, different weight vectors converged to different cluster centroids, thanks to the SWTA mechanism.

Fig. A.7. The illustration represents weight vectors (arrows) and data points (dots) within a data space. Fig. A.7s shows how a weight vector updates according to the SWTA equation when input is introduced. Fig. A.7b depicts the final positions of the weight vectors after multiple iterations, showing that each has converged to a distinct cluster centroid, guided by the SWTA mechanism.



(a) Initial position of weight and feature vectors.

(b) Evolution of weight and feature vectors after 20 training epochs.

(c) Final position of weight and feature vectors after 200 training epochs.

Fig. A.8. Visualization of feature vectors (blue) and weight vectors (orange) in latent space using our Hebbian SWTA-TSA pre-training approach in a real-world scenario (UNet backbone and GlaS dataset). Specifically, the feature vectors correspond to the flattened outputs of the UNet encoder, while the weight vectors are taken from the immediately subsequent upsampling layer. For clarity, only a subset of the feature and weight vectors is shown (more details about the visualization are in Appendix B). The three subfigures illustrate the evolution of weight and feature vectors across the training. Fig. A.8a shows the initial configuration, Fig. A.8b depicts the state after 20 training epochs, and Fig. A.8c presents the final state after 200 epochs. Two key observations can be made. First, the feature vectors shift over time as the encoder portion of the UNet backbone evolves during training. In particular, while the initial distribution is more uniform, data points tend to become more clustered as the network reorganizes information in the latent space. Second, the weight vectors tend to self-organize toward higher-density cluster regions; however, some neurons also cover data points in sparser regions. These may correspond to visual elements that occur less frequently in the images, exhibiting a form of outlier behavior..

when the weight vector $w_{i,1}$ converges to an eigenvector of the covariance matrix (and λ is the corresponding eigenvalue). This is exactly the definition of a principal component.

Another intuitive interpretation of the learning rule in Eq. (A.2) is the following: the term $\sum_{k=1}^{j} y_k, w_{i,k}$ can be viewed as a *reconstruction* of the observed input x_i, based on the current activations of neurons 1 through j. The weight update is proportional to the difference between the actual input x_i and this reconstruction. In other words, each neuron adjusts its weights in the direction that minimizes the *reconstruction error*. At the beginning of training, weight vectors are randomly initialized, resulting in a large reconstruction error. Consequently, the learning dynamics are initially fast, with strong updates driving the weight vectors toward the principal components of the data. As training progresses and the reconstruction improves, the updates become progressively smaller, leading to slower learning. This behavior is desirable, as it ensures stable weight dynamics and convergence of the neurons to an equilibrium configuration.

Appendix B. SWTA Latent Space Visualization

In this section, we visualize the feature vectors in the latent space and illustrate the dynamics of the weight vectors considering our Hebbian SWTA-TSA pre-training approach in a real-world scenario (see also Fig. A.8). To this end, we adopted a UNet backbone architecture—consistent with the rest of the methodology—and focused on the GlaS dataset, chosen for its complexity and richness in visual features.

Specifically, we trained the UNet using our unsupervised Hebbian pre-training method based on SWTA-TSA for 200 epochs, saving checkpoints at different stages to capture both model weights and data features. For each training sample, we extracted the latent feature representation from the encoder, flattening its output to obtain a feature vector. In parallel, we tracked the weight vectors of the first decoder layer, which directly receives the latent features as input. Checkpoints were recorded before training, at epoch 20, and at the final epoch, enabling us to analyze the evolution of both feature and weight vectors throughout the training process. Given the convolutional structure of the network, we partitioned the latent feature map from the UNet decoder into $k \times k$ patches, where k is the kernel size of the subsequent layer. Each patch was flattened to obtain a high-dimensional feature vector, and the collection of these vectors formed the dataset to be visualized. To reduce dimensionality, we applied PCA and used the resulting projection to map both feature and weight vectors into the same 2D space. Since plotting all data points and weights would result in a cluttered and hard-to-interpret figure, we selected a random subset of 500 feature vectors and 10 wt vectors, excluding outliers based on L2 norm to avoid numerical instability. These selected weight vectors were tracked across training, i.e., the same subset of feature and weight vectors was visualized at the three previously defined key stages (the initial condition at epoch 0, after 20 epochs, and at the final epoch).

The resulting visualizations are shown in Fig. A.8a, Fig. A.8b, and Fig. A.8c, respectively. These figures offer a compelling depiction of the learning process during Hebbian training. The latent feature representations undergo significant changes across epochs, driven by the evolving encoder configuration under synaptic plasticity. Initially, feature vectors appear roughly uniformly distributed; however, as training progresses, they gradually organize into clusters, indicating that the network is learning to structure the input information more effectively. Meanwhile, the weight vectors of the subsequent layer shift in response to the changing data distribution, attempting to follow the emerging centers of mass. By the end of training, we observe the formation of cluster-like structures—albeit with fuzzy boundaries—well captured by most of the weight vectors. The remaining vectors tend to occupy less densely populated regions of the latent space, likely corresponding to rare visual patterns in the dataset, which are nonetheless encoded by the network. Importantly, the latent feature vectors represent the network's internal encoding of the input images, while the weight vectors reflect its synaptic memory of the data distribution. This representation is initially inaccurate but becomes increasingly aligned with the actual data structure as training advances. In essence, the weight vectors adapt to form a quantized reconstruction of the input distribution, progressively improving in fidelity over time.

Appendix C. Implementation Details

In this section, we provide some implementation details. Our model is implemented using PyTorch, with all training and inference processes conducted on an NVIDIA DGX-A100. We also refer the reader to our repository available at https://tinyurl.com/hebbian-semantic-segmentation.

To ensure fair comparisons, we maintained consistent settings across all experiments. For the 2D datasets, data augmentation during training includes vertical and horizontal flips as well as rotations. Input images are resized to 128×128 for both training and inference. In contrast, for the LA dataset (Xiong et al., 2021), the same augmentation strategy is applied during training. Random patches of size $96 \times 96 \times 80$ are

extracted during training, while inference employs a sliding window approach with a 0.5 overlap ratio, using patches of the same size.

We used the SGD optimizer along with a multi-step learning rate scheduler to train the second stage of the two-stage approaches and the one-stage approaches. Conversely, we used the Adam optimizer to train the first unsupervised stage of the two-stage approaches. A multi-step learning rate scheduler is consistently applied in all cases. Specifically, the initial learning rate is set to 0.5 when using the SGD optimizer for the 2D datasets, and to 0.1 for the 3D LA dataset. For the Adam optimizer, we empirically observed better results with smaller initial learning rates: 0.001 for the 2D datasets and 0.0001 for the LA dataset.

We set the total number of training epochs to 200. In the first unsupervised stage, we always use the model snapshot from the final epoch to initialize the models for the second fine-tuning stage. Specifically, concerning Hebbian-based models, we noticed that they typically converge to a stable configuration of weights that do not change significantly after a few epochs. Empirically, we observed that model parameters across all datasets stabilized before reaching 200 epochs, which we therefore adopted as the training limit. Then, for the final evaluation on the test set, we select the best model based on DC performance on the validation split obtained during the fine-tuning stage. To ensure that our results are statistically relevant, we implement a 10-fold cross-validation protocol (5-fold for the LA dataset), varying the training, validation, and test splits.

Additionally, for the weight λ in the unsupervised loss functions of the competitor pseudo-labeling/consistency-based methods, we increase λ linearly with the number of epochs, following previous works, i.e., $\lambda = \lambda_{\max} \times \frac{epoch}{max\_epoch}$. We set $\lambda_{\max} = 5$, which we found to be the optimal value through a grid–search procedure.

Concerning the two-stage competitor approaches, we occasionally applied architectural modifications to better align each method's original task with the target task—semantic segmentation. Specifically, we added segmentation branches that were fine-tuned in the second stage and built on top of the features learned during the unsupervised stage. For SuperpixSSL, no changes were necessary, as its architecture was already designed for semantic segmentation. For VAE, the network's downsampling branch acts as the encoder, while the upsampling branch serves as the decoder. The decoder output is mapped through a linear layer to $256+256$ variables representing the mean and variance of a Gaussian distribution. A latent representation is sampled from this distribution and fed back into the decoder. During fine-tuning, the encoder parameters are initialized from the pretrained VAE model. The final layer used for input reconstruction is replaced with a new linear layer, initialized from scratch, which outputs the predicted segmentation map. For RAD-DINO (Perez-Garcia et al., 2025), we initialized the model using the pretrained image encoder from the original work and kept it frozen during the second stage. RAD-DINO is a foundation model, which by definition is trained on a massive amount of data. Therefore, we deemed it inappropriate to retrain it from scratch in the first stage using the relatively small dataset employed in this work. To produce segmentation maps, we added a decoding branch inspired by other two-stage methods. Each input image was mapped to a $37 \times 37 \times 768$ feature representation. The decoder consisted of a sequence of transpose convolutions: first, a layer with 768 input and 256 output channels (kernel size 3, stride 1), followed by ReLU and BatchNorm; then a layer with 256 to 128 channels (kernel size 3, stride 2), and another with 128 to 64 channels (kernel size 7, stride 3), both followed by ReLU and BatchNorm. The resulting features were upsampled via bilinear interpolation to 222×222 pixels, and finally passed through a transpose convolution with 64 input channels and one output channel per class (kernel size 3, stride 1) to generate the segmentation map. This decoder configuration was selected as it reflects the standard design based on transpose convolutions commonly used in semantic segmentation architectures. Training was performed independently on each dataset using the available labeled samples and the Adam optimizer with a learning rate of 0.01.

Finally, regarding the bio-inspired SNN model used for additional comparisons (Kim et al., 2022), the architectural and Leaky Integrate-and-Fire (LIF) parameters—such as membrane time constant and threshold—were set according to the original work. The model adopts a standard convolutional/transpose-convolutional backbone architecture, similar to the traditional backbone used in other baselines, but replaces conventional neuron models with spiking LIF counterparts (Abbott & van Vreeswijk, 1993). Training was performed using surrogate gradient methods, leveraging the available labeled samples in each scenario. The development of semi-supervised two-stage learning techniques based on spike-timing-dependent plasticity (STDP), analogous to the Hebbian strategies explored in this work, is left as future work, as it would require substantial theoretical advances in the modeling of STDP-based plasticity mechanisms.

Appendix D. Additional Experiments

In addition to the three public datasets described in Section 5.1, we also evaluate our approach on two further 2D biomedical image

Table D.9
Comparisons with SOTA on the OCT-CME dataset (Ahmed et al., 2022). **Bold** and *italic* indicate the best and second-best performance..

Labeled %	Method	DC (%) ↑	JI (%) ↑	95HD ↓	ASD ↓
1%	VAE	62.45 ± 5.55	46.69 ± 4.19	13.80 ± 3.58	5.03 ± 1.27
	SuperpixSSL	56.65 ± 3.92	41.39 ± 3.89	17.29 ± 3.88	5.05 ± 1.96
	RAD-DINO	49.92 ± 1.70	34.98 ± 1.30	20.72 ± 3.69	7.98 ± 2.07
	SpikingNN	51.33 ± 2.53	36.38 ± 1.47	17.44 ± 3.26	6.79 ± 2.07
	ADVENT	52.88 ± 2.56	37.84 ± 2.00	15.59 ± 1.67	6.56 ± 0.25
	CCT	62.14 ± 2.20	45.80 ± 2.53	13.42 ± 0.76	5.32 ± 0.22
	UAMT	60.79 ± 1.22	46.02 ± 0.72	13.13 ± 1.18	5.68 ± 1.09
	CPS	61.81 ± 1.05	44.27 ± 0.37	11.76 ± 1.63	4.95 ± 0.91
	URPC	*63.68 ± 4.25*	*48.98 ± 5.09*	**10.35 ± 2.44**	*3.62 ± 0.51*
	Ours	**64.04 ± 1.82**	**50.55 ± 1.27**	*10.58 ± 1.99*	**3.59 ± 0.47**
2%	VAE	63.77 ± 5.83	47.44 ± 5.64	11.05 ± 2.60	3.59 ± 1.42
	SuperpixSSL	61.98 ± 5.17	46.49 ± 5.48	14.03 ± 4.19	3.38 ± 1.35
	RAD-DINO	54.13 ± 0.62	38.32 ± 0.51	17.84 ± 0.42	6.37 ± 0.23
	SpikingNN	57.63 ± 0.67	41.99 ± 0.57	14.30 ± 1.02	5.47 ± 0.92
	ADVENT	58.76 ± 0.53	42.05 ± 1.01	13.99 ± 0.78	5.07 ± 0.73
	CCT	63.24 ± 0.74	47.79 ± 0.91	10.84 ± 0.37	4.78 ± 0.15
	UAMT	63.33 ± 0.25	46.98 ± 1.35	11.16 ± 1.05	4.88 ± 0.73
	CPS	65.01 ± 1.03	47.03 ± 0.38	10.50 ± 1.66	4.42 ± 0.78
	URPC	*65.93 ± 1.64*	*50.51 ± 2.14*	*10.03 ± 1.26*	**3.21 ± 0.77**
	Ours	**66.58 ± 1.31**	**51.71 ± 0.98**	**9.98 ± 0.64**	*3.10 ± 0.89*
5%	VAE	64.83 ± 0.87	49.81 ± 1.12	*9.11 ± 0.45*	3.56 ± 0.11
	SuperpixSSL	66.04 ± 1.92	49.34 ± 2.10	10.51 ± 1.34	*2.86 ± 0.50*
	RAD-DINO	56.55 ± 0.59	40.34 ± 0.50	15.12 ± 0.45	5.24 ± 0.29
	SpikingNN	61.37 ± 0.58	45.01 ± 1.02	12.07 ± 0.82	4.07 ± 0.19
	ADVENT	60.67 ± 0.98	43.98 ± 0.75	12.57 ± 0.34	4.23 ± 1.21
	CCT	66.47 ± 0.75	51.92 ± 0.96	9.73 ± 0.31	3.43 ± 0.07
	UAMT	65.98 ± 0.24	49.98 ± 1.01	10.11 ± 1.33	3.77 ± 0.48
	CPS	66.88 ± 1.01	48.98 ± 0.98	9.95 ± 0.76	3.02 ± 0.53
	URPC	*67.33 ± 1.45*	*53.10 ± 1.83*	9.60 ± 0.72	3.37 ± 0.20
	Ours	**67.99 ± 0.78**	**53.88 ± 0.28**	**8.88 ± 1.79**	**2.68 ± 0.67**
10%	VAE	67.84 ± 0.44	53.72 ± 0.59	8.98 ± 0.17	*2.25 ± 0.04*
	SuperpixSSL	67.58 ± 1.28	52.25 ± 1.58	8.34 ± 0.50	2.59 ± 0.14
	RAD-DINO	59.77 ± 0.37	43.13 ± 0.33	12.28 ± 0.53	4.44 ± 0.22
	SpikingNN	64.07 ± 1.71	48.84 ± 1.68	9.94 ± 0.89	3.10 ± 0.83
	ADVENT	62.54 ± 1.12	46.89 ± 0.76	11.00 ± 0.45	3.67 ± 0.12
	CCT	*68.69 ± 0.38*	54.88 ± 0.51	8.82 ± 0.32	2.49 ± 0.10
	UAMT	68.52 ± 0.81	52.66 ± 0.99	8.57 ± 1.22	2.92 ± 0.31
	CPS	67.37 ± 2.78	50.90 ± 2.11	9.54 ± 2.39	2.75 ± 0.81
	URPC	68.56 ± 1.97	**54.89 ± 2.72**	*8.34 ± 0.65*	2.56 ± 0.22
	Ours	**69.94 ± 3.33**	*53.94 ± 2.85*	**8.20 ± 2.49**	**2.20 ± 1.78**
20%	VAE	72.72 ± 0.22	56.91 ± 0.30	8.93 ± 0.26	2.21 ± 0.05
	SuperpixSSL	71.41 ± 0.44	53.15 ± 0.59	8.98 ± 0.20	2.21 ± 0.03
	RAD-DINO	62.67 ± 0.37	45.75 ± 0.34	11.52 ± 0.33	3.06 ± 0.18
	SpikingNN	70.39 ± 1.05	54.35 ± 0.15	8.88 ± 0.71	1.99 ± 0.23
	ADVENT	65.45 ± 1.21	49.98 ± 0.67	9.99 ± 0.54	2.77 ± 0.09
	CCT	*73.37 ± 0.34*	*58.33 ± 0.47*	7.51 ± 0.10	1.90 ± 0.04
	UAMT	71.35 ± 2.62	55.49 ± 3.15	8.53 ± 1.52	2.35 ± 0.59
	CPS	70.14 ± 0.84	54.02 ± 1.00	8.89 ± 0.64	2.41 ± 0.21
	URPC	73.21 ± 0.54	58.10 ± 0.58	**7.13 ± 0.94**	*1.85 ± 0.33*
	Ours	**74.00 ± 2.33**	**58.75 ± 2.94**	*7.16 ± 2.01*	**1.80 ± 0.57**

Table D.10
Comparisons with SOTA on the QaTa-COV19 dataset (Degerli et al., 2021). **Bold** and *italic* indicate the best and second-best performance. '—' indicates that the training process failed to reach convergence..

Labeled %	Method	DC (%) ↑	JI (%) ↑	95HD ↓	ASD ↓
1%	VAE	—	—	—	—
	SuperpixSSL	64.65 ± 2.92	49.39 ± 2.89	18.66 ± 1.01	8.44 ± 0.54
	RAD-DINO	59.92 ± 1.70	44.98 ± 1.30	19.72 ± 3.69	9.01 ± 0.98
	SpikingNN	59.26 ± 1.58	45.26 ± 1.28	18.95 ± 2.73	8.99 ± 1.73
	ADVENT	67.64 ± 2.59	51.84 ± 2.54	18.23 ± 1.42	8.44 ± 1.02
	CCT	69.32 ± 0.78	*53.16 ± 0.99*	17.02 ± 0.39	8.29 ± 0.15
	UAMT	68.93 ± 2.10	53.07 ± 2.47	16.98 ± 0.57	8.03 ± 0.36
	CPS	65.98 ± 0.81	50.24 ± 0.91	19.52 ± 0.39	8.11 ± 0.17
	URPC	*69.63 ± 0.81*	53.05 ± 1.00	**16.42 ± 0.81**	*7.99 ± 0.57*
	Ours	**70.02 ± 1.79**	**53.92 ± 2.32**	*16.72 ± 1.36*	**7.24 ± 0.34**
2%	VAE	—	—	—	—
	SuperpixSSL	67.24 ± 2.85	51.37 ± 2.95	18.10 ± 1.05	8.19 ± 0.52
	RAD-DINO	62.32 ± 1.65	46.78 ± 1.35	19.13 ± 3.60	8.74 ± 1.00
	SpikingNN	61.63 ± 1.60	47.07 ± 1.25	18.38 ± 2.70	8.72 ± 1.75
	ADVENT	70.35 ± 2.55	53.91 ± 2.60	17.68 ± 1.45	8.19 ± 1.00
	CCT	72.09 ± 0.92	*55.29 ± 1.07*	16.51 ± 0.51	8.04 ± 0.22
	UAMT	71.69 ± 2.36	55.29 ± 2.72	16.47 ± 0.69	7.79 ± 0.48
	CPS	68.62 ± 0.95	52.25 ± 1.08	18.93 ± 0.61	7.87 ± 0.29
	URPC	*72.42 ± 1.13*	55.17 ± 1.26	*15.93 ± 1.03*	*7.75 ± 0.69*
	Ours	**74.34 ± 0.98**	**59.19 ± 1.23**	**15.57 ± 0.49**	**6.58 ± 0.24**
5%	VAE	—	—	—	—
	SuperpixSSL	70.60 ± 2.67	53.94 ± 3.12	17.38 ± 1.21	7.86 ± 0.47
	RAD-DINO	66.68 ± 1.82	50.05 ± 1.49	17.79 ± 3.91	8.13 ± 0.87
	SpikingNN	65.94 ± 1.73	50.37 ± 1.42	17.09 ± 2.88	8.11 ± 1.62
	ADVENT	74.57 ± 2.71	57.14 ± 2.43	16.62 ± 1.38	7.70 ± 0.93
	CCT	76.42 ± 0.98	58.21 ± 1.14	15.52 ± 0.47	7.56 ± 0.26
	UAMT	76.00 ± 2.44	*58.61 ± 2.65*	15.48 ± 0.73	7.32 ± 0.45
	CPS	72.74 ± 1.02	55.39 ± 1.15	17.79 ± 0.66	7.40 ± 0.31
	URPC	*76.77 ± 1.21*	58.48 ± 1.33	*14.98 ± 1.09*	*7.29 ± 0.73*
	Ours	**79.25 ± 0.26**	**65.64 ± 0.36**	**13.62 ± 0.43**	**6.13 ± 0.12**
10%	VAE	—	—	—	—
	SuperpixSSL	74.84 ± 2.73	57.18 ± 3.05	16.34 ± 1.17	7.39 ± 0.50
	RAD-DINO	71.34 ± 1.96	53.55 ± 1.61	16.54 ± 4.02	7.56 ± 0.91
	SpikingNN	70.56 ± 1.88	53.90 ± 1.53	15.89 ± 2.97	7.54 ± 1.58
	ADVENT	78.54 ± 2.83	*64.71 ± 2.57*	15.29 ± 1.42	7.08 ± 0.89
	CCT	79.02 ± 1.78	61.36 ± 1.02	14.56 ± 0.89	6.99 ± 0.57
	UAMT	78.80 ± 1.37	62.45 ± 1.05	*14.05 ± 0.85*	*6.98 ± 0.79*
	CPS	77.39 ± 1.24	61.59 ± 0.73	14.75 ± 0.57	7.23 ± 0.83
	URPC	*79.12 ± 1.42*	62.98 ± 1.09	14.32 ± 0.88	7.11 ± 0.76
	Ours	**80.42 ± 0.25**	**66.26 ± 0.35**	**12.62 ± 0.26**	**5.88 ± 0.13**
20%	VAE	—	—	—	—
	SuperpixSSL	78.58 ± 2.87	62.75 ± 2.66	14.36 ± 1.26	7.02 ± 0.46
	RAD-DINO	74.91 ± 2.08	58.37 ± 1.52	15.38 ± 4.25	7.18 ± 0.87
	SpikingNN	74.09 ± 2.03	58.21 ± 1.67	14.94 ± 3.12	7.16 ± 1.45
	ADVENT	80.47 ± 2.96	67.89 ± 2.45	12.38 ± 1.53	6.23 ± 0.94
	CCT	80.17 ± 0.91	67.88 ± 1.08	12.54 ± 0.95	6.64 ± 0.52
	UAMT	80.74 ± 1.43	67.07 ± 1.12	12.07 ± 0.91	6.23 ± 0.85
	CPS	80.26 ± 1.33	67.13 ± 0.78	12.72 ± 0.61	*5.87 ± 0.89*
	URPC	*81.01 ± 0.45*	*68.02 ± 0.82*	*11.99 ± 0.52*	5.92 ± 0.67
	Ours	**81.08 ± 0.22**	**68.19 ± 0.29**	**11.34 ± 0.34**	**5.62 ± 0.12**

segmentation datasets, which involve distinct imaging modalities and segmentation tasks.

OCT-CME (Ahmed et al., 2022). The OCT-CME dataset contains Optical Coherence Tomography (OCT) images of diabetic macular edema patients, annotated for Cystoid Macular Edema (CME) segmentation. It includes manually labeled binary masks created using MATLAB tools to highlight CME regions pixel-wise.

QaTa-COV19 (Degerli et al., 2021). QaTa-COV19 is a large-scale chest X-ray (CXR) dataset developed for COVID-19 diagnosis, curated by Qatar and Tampere Universities. It contains over 120,000 CXR images, including 9258 confirmed COVID-19 cases, with 2951 segmentation masks highlighting infected lung regions.

Quantitative comparisons with state-of-the-art methods are reported in Table D.9 and Table D.10. In both cases, our approach outperforms previous works across all metrics and nearly all evaluated settings, confirming the findings presented in the main paper.

Appendix E. Additional Qualitative Results

This section presents additional qualitative results, complementing those shown in 5. Fig. E.9 reports qualitative predictions from all competing methods across the three main datasets (GlaS Sirinukunwattana et al., 2017, PH2 Mendonça et al., 2013, and HMEPS Mazziotti et al., 2021). For each dataset, we include qualitative examples corresponding to the 1%, 5%, and 20% labeled regimes, which reflect increasing levels of segmentation difficulty.

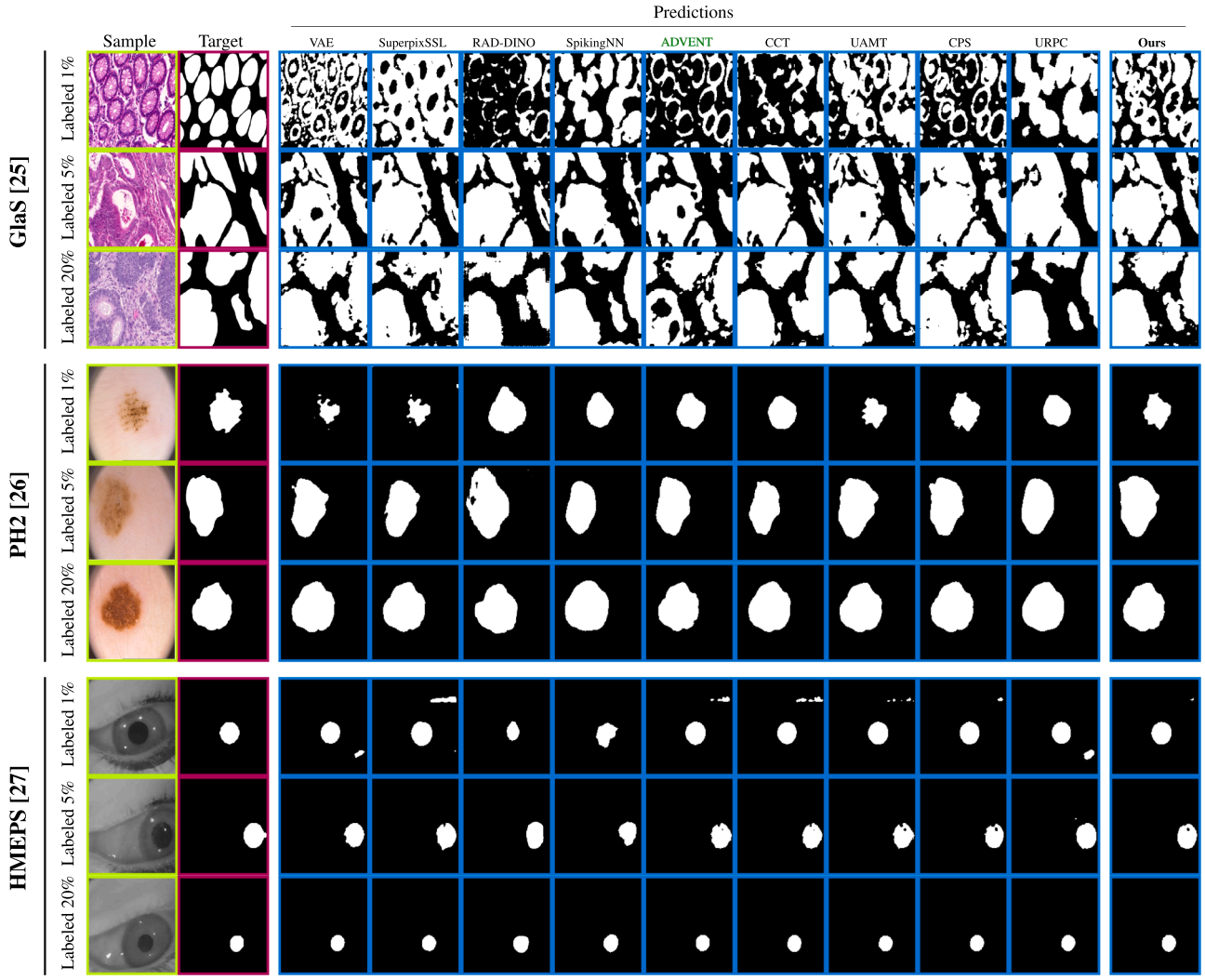


Fig. E.9. Qualitative comparisons across the GlaS (Sirinukunwattana et al., 2017), PH2 (Mendonça et al., 2013), and HMEPS (Mazziotti et al., 2021) datasets. For each dataset, we show examples from the 1%, 5%, and 20% labeled regimes, illustrating increasing segmentation difficulty. Each row includes the input, ground truth, predictions from all competing methods, and the output of the proposed approach.

References

Abbott, L. F., & van Vreeswijk, C. (1993). Asynchronous states in networks of pulse-coupled oscillators. *Physical Review E*, *48*(2), 1483.

Ahmed, Z., Panhwar, S. Q., Baqai, A., Umrani, F. A., Ahmed, M., & Khan, A. (2022). Deep learning based automated detection of intraretinal cystoid fluid. *International Journal of Imaging Systems and Technology*, *32*(3), 902–917. https://doi.org/10.1002/IMA.22662

Badar, A., Varma, A., Staniec, A., Gamal, M., Magdy, O., Iqbal, H., Arani, E., & Zonooz, B. (2021). Highlighting the importance of reducing research bias and carbon emissions in cnns. In *International conference of the italian association for artificial intelligence* (pp. 515–531). Springer.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615

Bahroun, Y., & Soltoggio, A. (2017). Online representation learning with single and multi-layer hebbian networks for image classification. In *Artificial neural networks and machine learning – ICANN 2017* (pp. 354–363). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-68600-4_41

Basak, H., Kundu, R., & Sarkar, R. (2022). Mfsnet: A multi focus segmentation network for skin lesion segmentation. *Pattern Recognition*, *128*(C). https://doi.org/10.1016/j.patcog.2022.108673

Becker, S., & Plumbley, M. (1996). Unsupervised neural network learning procedures for feature extraction and classification. *Applied Intelligence*, *6*(3), 185–203. https://doi.org/10.1007/bf00126625

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2006). Greedy layer-wise training of deep networks. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems*. MIT Press (*vol. 19*).

Bi, G.-q., & Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, *18*(24), 10464–10472.

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., & Wang, M. (2023). Swin-unet: Unet-like pure transformer for medical image segmentation. In *Computer vision – ECCV 2022 workshops* (pp. 205–218). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-25066-8_9

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International conference on computer vision, ICCV 2021, montreal, qc, canada, october 10–17, 2021* (pp. 9630–9640). IEEE. https://doi.org/10.1109/ICCV48922.2021.00951

Chen, L., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *CoRR*, *abs/1706.05587*.

Chen, R. J., Ding, T., Lu, M. Y., Williamson, D. F. K., Jaume, G., Song, A. H., Chen, B., Zhang, A., Shao, D., Shaban, M. et al. (2024). Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, *30*(3), 850–862.

Chen, X., Yuan, Y., Zeng, G., & Wang, J. (2021). Semi-supervised semantic segmentation with cross pseudo supervision. In *IEEE Conference on computer vision and pattern recognition, CVPR 2021, virtual, june 19–25, 2021* (pp. 2613–2622). Computer Vision Foundation / IEEE. https://doi.org/10.1109/CVPR46437.2021.00264

Ciampi, L., Carrara, F., Totaro, V., Mazziotti, R., Lupori, L., Santiago, C., Amato, G., Pizzorusso, T., & Gennaro, C. (2022). Learning to count biological structures with raters' uncertainty. *Medical Image Analysis*, *80*, 102500. https://doi.org/https://doi.org/10.1016/j.media.2022.102500

Ciampi, L., Lagani, G., Amato, G., & Falchi, F. (2024). A biologically-inspired approach to biomedical image segmentation. In *Computer vision – ECCV 2024 workshops* (pp. 158–171). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-91578-9_10

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3D u-net: Learning dense volumetric segmentation from sparse annotation. In *Medical image computing and computer-assisted intervention – MICCAI 2016* (pp. 424–432). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-46723-8_49

Degerli, A., Ahishali, M., Yamac, M., Kiranyaz, S., Chowdhury, M. E. H., Hameed, K., Hamid, T., Mazhar, R., & Gabbouj, M. (2021). COVID-19 Infection map generation and detection from chest x-ray images. *Health Information Science and Systems*, *9*(1), 15. https://doi.org/10.1007/S13755-021-00146-8

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Association for Computational Linguistics.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. arXiv:1603.07285.

Gansbeke, W. V., Vandenhende, S., Georgoulis, S., & Gool, L. V. (2021). Unsupervised semantic segmentation by contrasting object mask proposals. In *2021 IEEE/CVF International conference on computer vision, ICCV 2021, montreal, qc, canada, october 10–17, 2021* (pp. 10032–10042). IEEE. https://doi.org/10.1109/ICCV48922.2021.00990

Gerstner, W., & Kistler, W. M. (2002). Spiking neuron models: Single neurons, populations, plasticity. Cambridge university press.

Göltz, J., Kriener, L., Baumbach, A., Billaudelle, S., Breitwieser, O., Cramer, B., Dold, D., Kungl, A. F., Walter, S., Schemmel, J. et al. (2021). Fast and energy-efficient neuromorphic deep learning with first-spike times. *Nature Machine Intelligence*, *3*(9), 823–835.

Grossberg, S. (1991). Adaptive pattern classification and universal recoding, i: Parallel development and coding of neural feature detectors. In *Pattern Recognition by Self-Organizing Neural Networks*, pp. 203–232). The MIT Press. https://doi.org/10.7551/mitpress/5271.003.0008

Gupta, M., Modi, S. K., Zhang, H., Lee, J. H., & Lim, J. H. (2022). Is bio-inspired learning better than backprop? benchmarking bio learning vs. backprop. *CoRR*, *abs/2212.04614*. https://doi.org/10.48550/ARXIV.2212.04614

Hassabis, D., Kumaran, D., Summerfield, C., & Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. *Neuron*, *95*(2), 245–258.

Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., Roth, H. R., & Xu, D. (2022). Unetr: Transformers for 3d medical image segmentation. In *2022 IEEE/CVF Winter conference on applications of computer vision (WACV)* (pp. 1748–1758). https://doi.org/10.1109/WACV51458.2022.00181

Haykin, S. (2009). Neural networks and learning machines. (3rd ed.). Pearson.

Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J., & Maier-Hein, K. H. (2020). Nnu-net: A self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, *18*(2), 203–211. https://doi.org/10.1038/s41592-020-01008-z

Javed, F., He, Q., Davidson, L. E., Thornton, J. C., Albu, J., Boxt, L., Krasnow, N., Elia, M., Kang, P., Heshka, S. et al. (2010). Brain and high metabolic rate organ mass: contributions to resting energy expenditure beyond fat-free mass. *The American Journal of Clinical Nutrition*, *91*(4), 907–912.

Ji, X., Vedaldi, A., & Henriques, J. F. (2019). Invariant information clustering for unsupervised image classification and segmentation. In *2019 IEEE/CVF International conference on computer vision, ICCV 2019, Seoul, Korea (south), october 27, - november 2, 2019* (pp. 9864–9873). IEEE. https://doi.org/10.1109/ICCV.2019.00996

Journé, A., Rodriguez, H. G., Guo, Q., & Moraitis, T. (2022). Hebbian deep learning without feedback. arXiv preprint arXiv:2209.11883, .

Journé, A., Rodriguez, H. G., Guo, Q., & Moraitis, T. (2023). Hebbian deep learning without feedback. In *The eleventh international conference on learning representations*.

Karhunen, J., & Joutsensalo, J. (1995). Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Networks*, *8*(4), 549–562. https://doi.org/10.1016/0893-6080(94)00098-7

Karimi, A., Faez, K., & Nazari, S. (2023). Deu-net: Dual-encoder u-net for automated skin lesion segmentation. *IEEE Access*, *11*, 134804–134821. https://doi.org/10.1109/ACCESS.2023.3337528

Kim, Y., Chough, J., & Panda, P. (2022). Beyond classification: Directly training spiking neural networks for semantic segmentation. *Neuromorphic Computing and Engineering*, *2*(4), 044015. https://doi.org/10.1088/2634-4386/ac9b86

Kingma, D. P., Rezende, D. J., Mohamed, S., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Proceedings of the 27th international conference on neural information processing systems - volume 2* NIPS'14 (p. 3581–3589). Cambridge, MA, USA: MIT Press.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. In Y. Bengio, & Y. LeCun (Eds.), *2nd international conference on learning representations, ICLR 2014, banff, ab, canada, april 14–16, 2014, conference track proceedings*.

Krotov, D., & Hopfield, J. J. (2019). Unsupervised learning by competing hidden units. *Proceedings of the National Academy of Sciences*, *116*(16), 7723–7731. https://doi.org/10.1073/pnas.1820458116

Lagani, G., Falchi, F., Gennaro, C., & Amato, G. (2021). Hebbian semi-supervised learning in a sample efficiency setting. *Neural Networks*, *143*, 719–731. https://doi.org/https://doi.org/10.1016/j.neunet.2021.08.003

Lagani, G., Falchi, F., Gennaro, C., & Amato, G. (2022a). Comparing the performance of hebbian against backpropagation learning using convolutional neural networks. *Neural Computing and Applications*, *34*(8), 6503–6519. https://doi.org/10.1007/s00521-021-06701-4

Lagani, G., Falchi, F., Gennaro, C., & Amato, G. (2023). Synaptic plasticity models and bio-inspired unsupervised deep learning: A survey. *CoRR*, *abs/2307.16236*. https://doi.org/10.48550/ARXIV.2307.16236

Lagani, G., Falchi, F., Gennaro, C., Fassold, H., & Amato, G. (2024). Scalable bio-inspired training of deep neural networks with fasthebb. *Neurocomputing*, (p. 127867).

Lagani, G., Gennaro, C., Fassold, H., & Amato, G. (2022b). Fasthebb: Scaling hebbian training of deep neural networks to imagenet level. In *Similarity search and applications: 15th international conference, SISAP 2022, Bologna, Italy, october 5–7, 2022, proceedings* (pp. 251–264). Springer.

Lake, B. M., & Piantadosi, S. T. (2020). People infer recursive visual concepts from just a few examples. *Computational Brain & Behavior*, *3*, 54–65.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, *40*.

Larochelle, H., Bengio, Y., Louradour, J., & Lamblin, P. (2009). Exploring strategies for training deep neural networks. *Journal of Machine Learning Research : JMLR*, *10*, 1–40.

Lee, C., Sarwar, S. S., Panda, P., Srinivasan, G., & Roy, K. (2020). Enabling spike-based backpropagation for training deep neural network architectures. *Frontiers in Neuroscience*, *14*, 119.

Lei, Z., Yao, M., Hu, J., Luo, X., Lu, Y., Xu, B., & Li, G. (2025). Spike2former: Efficient spiking transformer for high-performance image segmentation. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 1364–1372). (*vol. 39*).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *IEEE Conference on computer vision and pattern recognition, CVPR 2015, Boston, MA, USA, june 7–12, 2015* (pp. 3431–3440). IEEE Computer Society. https://doi.org/10.1109/CVPR.2015.7298965

Luo, L. (2021). Architectures of neuronal circuits. *Science*, *373*(6559), eabg7285. https://doi.org/10.1126/science.abg7285

Luo, X., Chen, J., Song, T., & Wang, G. (2021). Semi-supervised medical image segmentation through dual-task consistency. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(10), 8801–8809. https://doi.org/10.1609/aaai.v35i10.17066

Luo, X., Wang, G., Liao, W., Chen, J., Song, T., Chen, Y., Zhang, S., Metaxas, D. N., & Zhang, S. (2022). Semi-supervised medical image segmentation via uncertainty rectified pyramid consistency. *Medical Image Analysis*, *80*, 102517. https://doi.org/https://doi.org/10.1016/j.media.2022.102517

Mazziotti, R., Carrara, F., Viglione, A., Leonardo, L., Luca, L. V., Alessandro, B., Giulia, R., Giulia, S., Giuseppe, A., & Tommaso, P. (2021). Human and Mouse Eyes for Pupil Semantic Segmentation. https://doi.org/10.5281/zenodo.4488164

Mendonça, T., Ferreira, P. M., Marques, J. S., Marcal, A. R. S., & Rozeira, J. (2013). Ph2 - a dermoscopic image database for research and benchmarking. In *2013 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC)* (pp. 5437–5440). https://doi.org/10.1109/EMBC.2013.6610779

Milletari, F., Navab, N., & Ahmadi, S. (2016). V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth international conference on 3d vision (3DV)* (pp. 565–571). Los Alamitos, CA, USA: IEEE Computer Society. https://doi.org/10.1109/3DV.2016.79

Moraitis, T., Toichkin, D., Journé, A., Chua, Y., & Guo, Q. (2022). Softhebb: Bayesian inference in unsupervised hebbian soft winner-take-all networks. *Neuromorphic Computing and Engineering*, *2*(4), 044017. https://doi.org/10.1088/2634-4386/aca710

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A. et al. (2023). Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, .

Ouali, Y., Hudelot, C., & Tami, M. (2020). Semi-supervised semantic segmentation with cross-consistency training. In *2020 IEEE/CVF Conference on computer vision and pattern recognition, CVPR 2020, Seattle, WA, USA, june 13–19, 2020* (pp. 12671–12681). Computer Vision Foundation / IEEE. https://doi.org/10.1109/CVPR42600.2020.01269

Ouyang, C., Biffi, C., Chen, C., Kart, T., Qiu, H., & Rueckert, D. (2020). Self-supervision with superpixels: Training few-shot medical image segmentation without annotation. In *Computer vision - ECCV 2020 - 16th European conference, Glasgow, UK, august 23–28, 2020, proceedings, part XXIX* (pp. 762–780). Springer (*vol. 12374*). Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-030-58526-6_45

Patel, K., Hunsberger, E., Batir, S., & Eliasmith, C. (2021). A spiking neural network for image segmentation. arXiv preprint arXiv:2106.08921, .

Pehlevan, C., Hu, T., & Chklovskii, D. B. (2015). A hebbian/anti-Hebbian neural network for linear subspace learning: A derivation from multidimensional scaling of streaming data. *Neural Computation*, *27*(7), 1461–1495. https://doi.org/10.1162/NECO_a_00745

Perez-Garcia, F., Sharma, H., Bond-Taylor, S., Bouzid, K., Salvatelli, V., Ilse, M., Bannur, S., Castro, D. C., Schwaighofer, A., Lungren, M. P. et al. (2025). Exploring scalable medical image encoders beyond text supervision. *Nature Machine Intelligence*, *7*(1), 119–130.

Roh, Y., Heo, G., & Whang, S. E. (2021). A survey on data collection for machine learning: A big data - AI integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, *33*(4), 1328–1347. https://doi.org/10.1109/TKDE.2019.2946162

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention – MICCAI 2015* (pp. 234–241). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28

Rumelhart, D. E., & Zipser, D. (1985). Feature discovery by competitive learning*. *Cognitive Science*, *9*(1), 75–112. https://doi.org/10.1207/s15516709cog0901_5

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, *2*(6), 459–473. https://doi.org/10.1016/0893-6080(89)90044-0

Schuman, C. D., Kulkarni, S. R., Parsa, M., Mitchell, J. P., Date, P., & Kay, B. (2022). Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, *2*(1), 10–19.

Shrestha, A., Fang, H., Mei, Z., Rider, D. P., Wu, Q., & Qiu, Q. (2022). A survey on neuromorphic computing: Models and hardware. *IEEE Circuits and Systems Magazine*, *22*(2), 6–35.

Sirinukunwattana, K., Pluim, J. P. W., Chen, H., Qi, X., Heng, P.-A., Guo, Y. B., Wang, L. Y., Matuszewski, B. J., Bruni, E., Sanchez, U., Böhm, A., Ronneberger, O., Cheikh,

B. B., Racoceanu, D., Kainz, P., Pfeiffer, M., Urschler, M., Snead, D. R. J., & Rajpoot, N. M. (2017). Gland segmentation in colon histology images: The glas challenge contest. *Medical Image Analysis*, *35*, 489–502. https://doi.org/https://doi.org/10.1016/j.media.2016.08.008

Valanarasu, J. M. J., Oza, P., Hacihaliloglu, I., & Patel, V. M. (2021). Medical transformer: Gated axial-attention for medical image segmentation. In *Medical image computing and computer assisted intervention – MICCAI 2021* (pp. 36–46). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-87193-2_4

Valanarasu, J. M. J., Sindagi, V. A., Hacihaliloglu, I., & Patel, V. M. (2022). Kiu-net: Overcomplete convolutional architectures for biomedical image and volumetric segmentation. *IEEE Transactions on Medical Imaging*, *41*(4), 965–976. https://doi.org/10.1109/TMI.2021.3130469

Vu, T., Jain, H., Bucher, M., Cord, M., & Pérez, P. (2019). ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *IEEE Conference on computer vision and pattern recognition, CVPR 2019, Long Beach, CA, USA, june 16–20, 2019* (pp. 2517–2526). Computer Vision Foundation / IEEE. https://doi.org/10.1109/CVPR.2019.00262

Wang, H., He, Z., Wang, T., He, J., Zhou, X., Wang, Y., Liu, L., Wu, N., Tian, M., & Shi, C. (2022). Triplebrain: A compact neuromorphic hardware core with fast on-chip self-organizing and reinforcement spike-timing dependent plasticity. *IEEE Transactions on Biomedical Circuits and Systems*, *16*(4), 636–650.

Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., & Shi, L. (2019). Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 1311–1318). (*vol. 33*).

Xie, Y., Zhang, J., Shen, C., & Xia, Y. (2021). Cotr: Efficiently bridging CNN and transformer for 3d medical image segmentation. In *Medical image computing and computer assisted intervention – MICCAI 2021* (pp. 171–180). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-87199-4_16

Xiong, Z., Xia, Q., Hu, Z., Huang, N., Bian, C., Zheng, Y., Vesal, S., Ravikumar, N., Maier, A., Yang, X., Heng, P.-A., Ni, D., Li, C., Tong, Q., Si, W., Puybareau, E., Khoudli, Y., Géraud, T., Chen, C., Bai, W., Rueckert, D., Xu, L., Zhuang, X., Luo, X., Jia, S., Sermesant, M., Liu, Y., Wang, K., Borra, D., Masci, A., Corsi, C., de Vente, C., Veta, M., Karim, R., Preetha, C. J., Engelhardt, S., Qiao, M., Wang, Y., Tao, Q., Nuñez-Garcia, M., Camara, O., Savioli, N., Lamata, P., & Zhao, J. (2021). A global benchmark of algorithms for segmenting the left atrium from late gadolinium-enhanced cardiac magnetic resonance imaging. *Medical Image Analysis*, *67*, 101832. https://doi.org/https://doi.org/10.1016/j.media.2020.101832

Xu, H., Usuyama, N., Bagga, J., Zhang, S., Rao, R., Naumann, T., Wong, C., Gero, Z., González, J., Gu, Y. et al. (2024). A whole-slide foundation model for digital pathology from real-world data. *Nature*, *630*(8015), 181–188.

Yu, L., Wang, S., Li, X., Fu, C.-W., & Heng, P.-A. (2019). Uncertainty-aware self-ensembling model for semi-supervised 3d left atrium segmentation. In *Medical image computing and computer assisted intervention – MICCAI 2019* (pp. 605–613). Cham: Springer International Publishing.

Yue, Y., Baltes, M., Abuhajar, N., Sun, T., Karanth, A., Smith, C. D., Bihl, T., & Liu, J. (2023). Spiking neural networks fine-tuning for brain image segmentation. *Frontiers in Neuroscience*, *17*, 1267639.

Zhang, Y., Lee, K., & Lee, H. (2016). Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. In *Proceedings of the 33rd international conference on international conference on machine learning - volume 48* ICML'16 (p. 612–621). JMLR.org.

Zhou, S., Li, X., Chen, Y., Chandrasekaran, S. T., & Sanyal, A. (2021). Temporal-coded deep spiking neural network with easy training and robust performance. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 11143–11151). (*vol. 35*).