Journal Pre-proof

A Positive/Unlabeled Approach for the Segmentation of Medical Sequences using Point-Wise Supervision

Laurent Lejeune, Raphael Sznitman

PII: S1361-8415(21)00231-0
DOI: https://doi.org/10.1016/j.media.2021.102185
Reference: MEDIMA 102185

To appear in: *Medical Image Analysis*

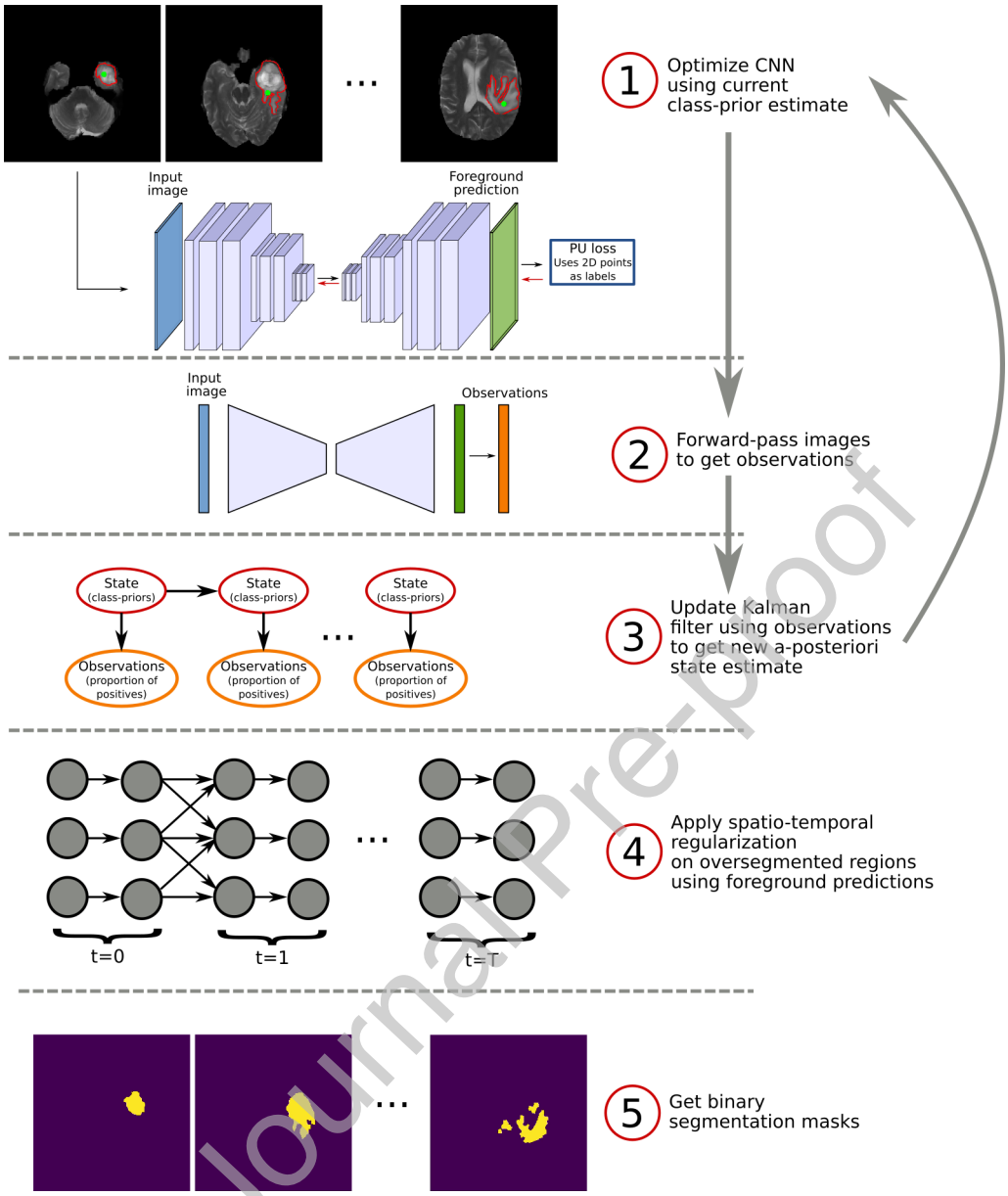
Received date: 9 April 2021
Revised date: 25 June 2021
Accepted date: 16 July 2021

Please cite this article as: Laurent Lejeune, Raphael Sznitman, A Positive/Unlabeled Approach for the Segmentation of Medical Sequences using Point-Wise Supervision, *Medical Image Analysis* (2021), doi: https://doi.org/10.1016/j.media.2021.102185

Highlights

- Ground-truth annotation are necessary to train state-of-the-art Machine Learning models.
- We annotate video and volumetric sequences using a maximum of one 2D point per frame.
- No constraints on appearance, shape, and motion/displacement of object of interest.
- Substantial improvement over state-of-the-art methods on various imaging modalities: surgical tool, slitlamp videos, brain MRI, and CT scans of inner ear.

Graphical Abstract



Medical Image Analysis (2021)

Contents lists available at ScienceDirect

Medical Image Analysis

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

A Positive/Unlabeled Approach for the Segmentation of Medical Sequences using Point-Wise Supervision

Laurent Lejeune, Raphael Sznitman

Artificial Intelligence in Medical Imaging, ARTORG Center, University of Bern, Murtenstrasse 50, 3008 Bern, Switzerland

ARTICLE INFO

Article history:

Transductive Learning, Positive-Unlabeled learning, Semantic segmentation, Point-wise supervision

ABSTRACT

The ability to quickly annotate medical imaging data plays a critical role in training deep learning frameworks for segmentation. Doing so for image volumes or video sequences is even more pressing as annotating these is particularly burdensome. To alleviate this problem, this work proposes a new method to efficiently segment medical imaging volumes or videos using point-wise annotations only. This allows annotations to be collected extremely quickly and remains applicable to numerous segmentation tasks. Our approach trains a deep learning model using an appropriate Positive/Unlabeled objective function using sparse point-wise annotations. While most methods of this kind assume that the proportion of positive samples in the data is known a-priori, we introduce a novel self-supervised method to estimate this prior efficiently by combining a Bayesian estimation framework and new stopping criteria. Our method iteratively estimates appropriate class priors and yields high segmentation quality for a variety of object types and imaging modalities. In addition, by leveraging a spatio-temporal tracking framework, we regularize our predictions by leveraging the complete data volume. We show experimentally that our approach outperforms state-of-the-art methods tailored to the same problem.

1. Introduction

Modern machine learning methods for semantic segmentation have shown excellent performances in recent years across numerous medical domains. These advances have been spearheaded by new deep learning based approaches that leverage large amounts of images and annotations. Unfortunately however, generating substantial quantities of segmentation annotations for medical imaging remains extremely burdensome. This is in part due to the need for domain specific knowledge in clinical applications, but also due to the fact that many medical imaging modalities are often 3-dimensional (*e.g.,* CT, MRI) or video based (*e.g.,* endoscopy, microscopy, etc.). The latter point greatly increases the necessary time and effort to manually segment even just a few volumes or videos.

To overcome this important bottleneck, semi-supervised learning in medical imaging has been an active research area. Sub-domains have included, active learning (Sener and Savarese, 2018; Konyushkova et al., 2015), self-supervised methods (Chen et al., 2019; Jamaludin et al., 2017), or crowdsourcing based annotating (Salvador et al., 2013; Heim et al., 2018), all of which have variants tailored for segmentation tasks. Broadly speaking, the core idea in each of these is to maximize the value of individual manually generated annotations for subsequent training of segmentation models. One subtype of semi-supervised methods, known as *transductive learn-*

e-mail: raphael.sznitman@artorg.unibe.ch (Raphael Sznitman)

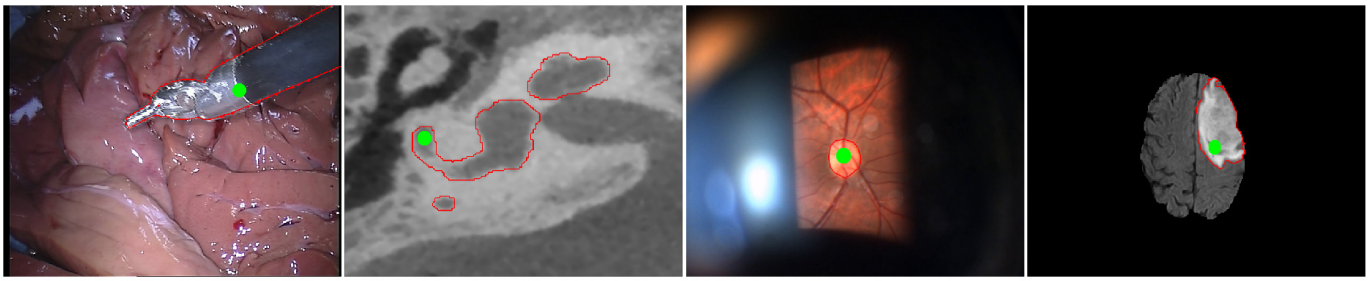


Fig. 1. Examples of image frames in different imaging modalities with different objects of interest to segment. In each example, a 2D point annotation is show in green and the complete pixel-wise groundtruth segmentation is shown in red. Applications shown are (from left to right): Video frame of a surgical instrument during minimally invasive surgery, a single slice from a CT scan depicting a human cochlea, video frame from a slitlamp examination of the optic nerve, brain slice from an MRI scan showing a tumor.

ing, considers the case where all data points are initially given and the labels of certain data points must be inferred from a subset of known annotated data points. As such, it is closely related to belief propagation (Knbelreiter et al., 2020) and unsupervised learning.

We thus consider inferring a complete segmentations of a given 3D volume or video sequence from limited user-provided annotations, as a transductive learning problem. That is, we wish to segment all pixels of a given data volume, whilst only having access to a few partial locations being annotated. Unlike traditional graphcut based methods (Boykov et al., 2001) that do so by using both positive and negative samples (*i.e.* PN learning), our focus is on cases where only positive samples are accessible. With some application-specific solutions previously developed (Vilariño et al., 2007; Khosravan et al., 2017), we follow the line of (Bearman et al., 2016; Lejeune et al., 2017, 2018), aiming for agnostic solutions capable of working for different unknown object of interest (shape, appearance, motion, etc.), as well as different imaging modality (MRI, CT-scan, video, etc.). Fig. 1 illustrates a number of different such scenarios and highlights the broad range appearances and settings considered here.

To further reduce the annotation burden, we choose to make use of sparse point-wise annotations to indicate pixels that belong to the object of interest in given data volumes. As observed in Bearman et al. (2016), point-wise supervision are extremely easy to collect and very reliable. While a manual segmentation may take on average 239 seconds for a single PASCAL VOC image, the corresponding multi-class point-wise annotations only requires 22 seconds. In this work, we further restrict the point-wise annotations to only be available on the object of interest, and not on the background regions of the image. Additionally, point-wise annotations can be provided by manual clicks (Ferreira et al., 2012; Bearman et al., 2016), or using a gaze tracker (Vilariño et al., 2007; Khosravan et al., 2017; Lejeune et al., 2017, 2018). While this convenience and speed gain comes at the cost of extremely sparse annotations in contrast to full semantic segmentations, methods to generate full segmentations in this setting have shown some promising performances in medical imaging. The present work follows this specific line of research.

Our goal is thus to generate a sequence of binary segmentation masks for an object of interest present in a volume or video using point-wise annotations that only specify the object. That is, the annotations only provide explicit information as to the location of the object of interest and no information regarding the background is known. To do this, we would ideally like to train a function to learn from annotated locations, take into account unlabeled locations and infer their labels. While it would appear that using neural networks to do so would be the obvious choice, doing so is challenging because positive samples are given by annotations (*i.e.,* locations on the object of interest throughout the data), yet no explicit negative samples are available. Instead, a large unlabeled set of samples is accessible without knowing which of these is positive or negative. This problem setting is known as P(ositive)-U(nlabeled) learning (Li and Liu, 2003, 2005; Du Plessis and Sugiyama, 2014; Plessis et al., 2015) and lies at the heart of this work.

We thus introduce a novel PU learning method that allows for high segmentation performances from point-wise annotations in a transductive learning setting. Our approach leverages a non-negative unbiased risk estimator to infer the likelihood of the object presence throughout the data. Specifically, we use this estimator as an objective function to train a deep learning model in a transductive setup. As the estimator requires accurate class-priors to be effective, we introduce a self-supervised strategy to estimate the proportion of positive samples on each frame using a recursive bayesian approach within our training procedure. Our method has the benefit of only needing a single upper-bound initialization value while allowing per-frame estimates to be computed. We further combine the latter estimates with a multi-path tracking framework that explicitly leverages the spatio-temporal relations of over-segmented regions. This allows the output of our model to be regularized throughout the data volume. We show experimentally that our pipeline brings important performance gains over state-of-the-art methods across a broad range of image modalities and object types.

2. Related Works

In this section we provide an overview of some of the most relevant related works to the method presented here.

Positive/Unlabeled learning considers the learning setting where only a subset of the positive samples are labeled, while the unlabeled set contains both positive and negative samples.

Early methods focused on iteratively sampling confident negatives from the unlabeled set using a classifier, while re-training the same classifier using these (Li and Liu, 2003; Liu et al., 2003; Li and Liu, 2005). In Lee and Liu (2003), the authors propose a reweighing scheme applied to the unlabeled samples, which allows the use of traditional supervised classifiers. As the latter approach heavily relies on appropriate weights, Elkan and Noto (2008) instead chose to duplicate unlabeled samples into positive and negative samples with complementary weights, an approach called unbiased PU learning. More recently, a general-purpose unbiased risk estimator for PU learning was presented by Plessis et al. (2015) which allows for convex optimization in the PU setting. As a follow-up to the latter, Kiryo et al. (2017) noted that modern expressive models, such as Deep Neural Networks, induce negative empirical risks through overfitting of the positives, which they propose to fix by introducing a non-negative unbiased risk estimator. We detail this method in the next section as we build directly from this method. Beyond the works mentioned here, we invite the reader to a more complete review of the topic in Bekker and Davis (2020), as many new methods have now focused on specific settings such as biased-negative samples (Hsieh et al., 2019) or temporal shifts in the positive class (Akujuobi et al., 2020).

Class-prior estimation is tightly related to the state-of-the-art PU learning approaches that design risk estimators relying on knowing or estimating the density of positive and negative samples. In Du Plessis and Sugiyama (2014), the authors suggest to partially match the class-conditional densities of the positive class to the input samples using the Pearson divergence criteria. In Christoffel et al. (2016), the same approach is improved by considering a general divergence criteria along with a L_1 distance regularization, which diminishes the problem of over-estimation. In Bekker and Davis (2018), a tree induction scheme is introduced to estimate the probability of positive samples as a proxy task (Scott and Blanchard, 2009). Similar to the self-supervised estimation of class-priors proposed in the present work, Kato et al. (2018) combines the non-negative unbiased risk of Kiryo et al. (2017) with an iterative update of class-priors. In particular, they devise an update rule inspired by the Expectation-Maximization algorithm and iterate until convergence.

Point-wise supervision was first applied in the context of medical image analysis in Vilariño et al. (2007), where a Support Vector Machine was used to classify patches. Using a graph approach, Khosravan et al. (2017) constructed saliency maps as the input of a Random-Walker to segment CT volumes. More generally, Bearman et al. (2016) train a CNN using a loss function that includes an object-prior defined by the user-provided points. The most relevant methods to the present work are those of Lejeune et al. (2017) and Lejeune et al. (2018). In the former, a classifier is learned to segment various kinds of medical image volumes and videos in a PU setting using a loss function that leverages the uncertainties of unlabeled samples. As a follow-up, Lejeune et al. (2018), formulated the same problem as a multi-path optimization problem. In particular, a simple object model formulated in a PU setting provides costs of selecting superpixels. The multi-path framework showed good performances in inferring structures that were non-concave (*e.g.,* a doughnut shape).

3. Methods

The goal of our method is to generate a segmentation mask for an object of interest in each frame of a single image volume or video sequence using only point-wise annotations and without knowing the image type or object of interest beforehand.

To do so, we propose a novel approach within the Positive-Unlabeled learning setting, that learns the segmentation of the object identified by the point-wise annotations. Our method makes use of the non-negative risk estimator introduced in Kiryo et al. (2017), which heavily relies on knowing the proportion of positive samples in the data. While unavailable in our setting, we introduce a novel self-supervised method to estimate this key value via an iterative learning procedure within a Bayesian estimation framework. We then devise a stopping condition to halt training at an appropriate point. Last, a spatio-temporal tracking framework is applied to regularize the output of our approach over the complete volume information.

We describe our approach now in more detail. In the following subsection, we introduce the Positive-Unlabeled learning framework and its non-negative risk estimator. We then describe in Sec. 3.2 our self-supervised approach to learn effective class priors. Last, we detail how we leverage the spatio-temporal regularizer, and provide our implementation details in Sec. 3.3 and Sec. 3.4, respectively.

3.1. Non-negative Positive-Unlabeled learning

We first briefly introduce and formulate Non-negative Positive-Unlabeled learning Kiryo et al. (2017) in the context of semantic segmentation.

Traditional supervised learning, which we denote Positive/Negative learning (PN), looks to build a model $f_\theta : I \mapsto [0;1]^{W\times H}$, where θ is a set of model parameters, I is the input image with width W and height H. Letting $\boldsymbol{\mathcal{I}} = \{\mathcal{I}^i\}_{i=1}^N$ be the set of N input images corresponding to an image volume or video sequence, each image $\mathcal{I}^i$ is composed of pixels, $\mathcal{X}^i = \mathcal{X}^i_p \cup \mathcal{X}^i_n$, where $\mathcal{X}^i_p$ and $\mathcal{X}^i_n$ denote the positive and negative pixels, respectively. We denote $\pi^i \in (0,1)$ as the proportion of positive pixels in image i, and $\boldsymbol{\pi} = \{\pi^i\}_{i=1}^N$ as the set of such proportions over all frames.

Training f to optimize θ can then be computed by minimizing the empirical risk of the form,

$$R_{pn} = \sum_{i=1}^{N}\left[\frac{\pi^i}{|\mathcal{X}^i_p|}\sum_{x\in\mathcal{X}^i_p}\ell^+(f_\theta(x)) + \frac{1-\pi^i}{|\mathcal{X}_n|}\sum_{x\in\mathcal{X}^i_n}\ell^-(f_\theta(x))\right]. \quad (1)$$

A popular choice for ℓ is the logistic loss, for which $\ell^+(z) = \log(1+e^{-z})$, $\ell^-(z) = \log(1+e^{z})$ are the positive and negative entropy loss terms, respectively. In which case, Eq. (1) is the Balanced Cross-Entropy loss (BBCE).

Conversely, computing Eq. (1) is infeasable in a PU settting, as neither $\boldsymbol{\pi}$ nor $\boldsymbol{\mathcal{X}}_n$ are known in advance. Instead, we have a set of unlabeled samples $\boldsymbol{\mathcal{X}}_u$ that contain both positives and

negatives. As suggested in Plessis et al. (2015), the negative risk (*i.e.,* the second term of Eq. (1)) can however be re-written in terms of $\boldsymbol{\mathcal{X}}_p$ and $\boldsymbol{\mathcal{X}}_u$ as,

$$R_{pu} = \sum_{i=1}^{N}\Bigg[\frac{\pi^i}{|\mathcal{X}_p^i|}\sum_{x\in\mathcal{X}_p^i}\ell^+(f_\theta(x))+ \left(\frac{1}{|\mathcal{X}_u^i|}\sum_{x\in\mathcal{X}_u^i}\ell^-(f_\theta(x)) - \frac{\pi^i}{|\mathcal{X}_p^i|}\sum_{x\in\mathcal{X}_p^i}\ell^-(f_\theta(x))\right)\Bigg]. \quad (2)$$

This is achieved by observing that $p(x) = \pi p(x|Y=1) + (1-\pi)p(x|Y=-1)$ and that the negative risk can be expressed as $(1-\pi)\mathbb{E}_{X\sim p(x|Y=-1)}\left[\ell^-(f_\theta(X))\right] = \mathbb{E}_{X\sim p(x)}\left[\ell^-(f_\theta(X))\right] - \pi\mathbb{E}_{X\sim p(x|Y=+1)}\left[\ell^-(f_\theta(X))\right]$. In the case of expressive models such as Neural Networks, minimizing the objective of Eq. (2) using stochastic gradient descent on mini-batches of samples tends to overfit to the training data, by driving the negative risk, (*i.e.,* the bottom term of Eq. (2)) to be negative.

To circumvent this, Kiryo et al. (2017) proposed to perform gradient ascent when the negative risk of a mini-batch is negative using the following negative risk,

$$R_i^- = \sum_{i=1}^{N}\left(\frac{1}{|\mathcal{X}_u^i|}\sum_{x\in\mathcal{X}_u^i}\ell^-(f_\theta(x)) - \frac{\pi^i}{|\mathcal{X}_p^i|}\sum_{x\in\mathcal{X}_p^i}\ell^-(f_\theta(x))\right). \quad (3)$$

Thus the complete training procedure for the PU setting with deep neural networks is described in Alg. 1. Specifically, when $R_i^- < 0$, gradient ascent is performed by setting the gradient to $-\nabla_\theta R_i^-$.

Algorithm 1 Non-negative PU learning

Input: f_θ: Prediction model
$\mathcal{I}$: Set of images
$\boldsymbol{\mathcal{X}_p}, \boldsymbol{\mathcal{X}_u}$: Positive and unlabeled samples
$\boldsymbol{\pi}$: Set of class priors
T: Number of epochs

for epoch $\leftarrow 1$ to T **do**
  Shuffle dataset into N_b batches
  for $i \leftarrow 1$ to N_b **do**
    Sample next batch to get $\mathcal{I}^i, \mathcal{X}_p^i, \mathcal{X}_u^i, \boldsymbol{\pi}^i$
    Forward pass $\mathcal{I}^i$ in f_θ
    Compute risks as in Eq. 2 and 3
    if $R_i^- < 0$ **then**
      Do gradient ascent along $\nabla_\theta R_i^-$
    else
      Do gradient descent along $\nabla_\theta R_{pu}$
    end if
  end for
end for

Critically, $\boldsymbol{\pi}$ plays an important role in Eq. (2) and Eq. (3). While Kiryo et al. (2017) assumes that the class prior is known and constant across all the data, this is not the case in many applications, including the one at the heart of this work. In particular, the class prior here is specific to each frame of the image data available (*i.e.,* π^i) as each frame may have different numbers of positives (*e.g.,* the object may appear bigger or smaller). In addition, we show in our experimental section that setting the class prior in naive ways leads to low performance levels. In the subsequent subsection, we hence introduce a novel approach to overcome this limitation.

3.2. Self-supervised Class-priors Estimation

Instead of fixing the values of $\boldsymbol{\pi}$ before training as in Kiryo et al. (2017), we instead propose to refine all values iteratively during training. Our approach, which we refer to as Self-Supervised Non-Negative PU Learning (**SSnnPU**), will start with a large-upper bound for π_i and will progressively reduce the estimates at each epoch of our training scheme until a stopping criterion is reached. That is, we will optimize the function f_θ one epoch at a time using Alg. 1, and then use the intermediary model to help estimate the class priors.

However, deriving class prior estimates from partial models (*i.e.,* trained with few epochs) yields very noisy estimates with large variances. Hence, we propose to use a Bayesian framework to estimate the class priors in a recursive fashion by establishing a state space and observation model and inferring the class priors. This is motivated by the fact that most PU learning methods developed so far (Bekker and Davis, 2020) rely on the Selected Completely At Random (SCAR) assumption to model positive samples. In our case, this is not the case however as positive samples are highly correlated given that they correspond to pixels in images and thus have strong correlations with other positive samples. We now describe our approach in more detail by first formalizing the state and observation models, and we describe our recursive Bayesian estimate and stopping conditions thereafter. Our final training algorithm is summarized in Alg. 2.

3.2.1. State and Observation models

Recall $\boldsymbol{\pi}_k = \{\pi_k^i\}_{i=1}^N$ to be the true class prior of a sequence of N frames. While we wish to know $\boldsymbol{\pi}$, our method will compute values $\hat{\boldsymbol{\pi}} = \{\hat{\pi}^i\}_{i=1}^N$ as the best approximation to $\boldsymbol{\pi}$. At the same time, our prediction model after k training epochs, denoted f_{θ_k}, can also provide partial information to the value of $\boldsymbol{\pi}$. Specifically, by evaluating f_{θ_k} on all samples $x \in \mathcal{X}^i$, we can estimate a noisy observation of π^i by computing the expected value over f_{θ_k},

$$\rho_k^i = \mathbb{E}_{x\in\mathcal{X}^i}[f_{\theta_k}(x)^\gamma]. \quad (4)$$

Here, $\gamma > 1$ is a correction factor that mitigates variations in the expectation at different epoch values. That is, we wish that our prediction model slightly over-segment the object of interest so to over-estimate the frequencies of positives. This is because we wish to progressively decrease $\hat{\boldsymbol{\pi}}$ from its initial value $\hat{\pi}_0$ by using $\boldsymbol{\rho}_k$ as observations.

To do this, we denote $\boldsymbol{\pi_k}$ and $\hat{\boldsymbol{\pi}}_{\boldsymbol{k}}$ to be true and inferred class priors after epoch k. While the value $\boldsymbol{\pi_k}$ is the same for all values of k, we include this notation at this stage to define the following linear state observation model we will use to infer $\hat{\boldsymbol{\pi}}_{\boldsymbol{k}}$,

$$\boldsymbol{\pi}_{k+1} = g(\boldsymbol{\pi}_k, L) - u_k\mathbf{1}_N + \mathcal{N}(0, Q), \quad (5)$$

Algorithm 2 Self-supervised Non-Negative PU Learning

Input: $\hat{\pi}_0$: Upper-bound on class-priors
T: Number of epochs
f_θ: Foreground prediction model
Output: Optimal estimate of class-prior $\hat{\boldsymbol{\pi}}^*$

$k = 0$
$\hat{\pi}_0^i = \pi_{max}, \forall i$
while Stopping condition not satisfied **do**
  Optimize f_θ for 1 epoch using Alg. 1 and $\hat{\boldsymbol{\pi}}_k$
  Compute observations $\boldsymbol{\rho}_k$ as in Eq. (4)
  Clip $\boldsymbol{\rho}_k$ to $[0, \hat{\pi}_0]$
  Compute $\hat{\boldsymbol{\pi}}_{k+1}$ using UKF with $\boldsymbol{\rho}_k$ and $\hat{\boldsymbol{\pi}}_k$
  $k \leftarrow k + 1$
end while

where $\boldsymbol{\rho}_k \sim \mathcal{N}(\boldsymbol{\pi}_k, R)$, $\boldsymbol{\pi}_0 \sim \mathcal{N}(\hat{\pi}_0, S)$, and Q, R, and S are the transition, observation, and initial covariance matrices, respectively. The function $g(\cdot, L)$ is a moving average filter of length L with a Hanning window to impose a frame-wise smoothness. For convenience, we write $\mathbf{1}_N$ for a vector of length N taking values of 1, and the term u_k is the control input,

$$u_k = u_0 + (u_T - u_0)\frac{k}{T}, \tag{6}$$

where u_0 and u_T are two scalars such that $u_T > u_0$. The control input therefore induces a downward acceleration on the states and imposes a "sweeping" effect on the latter, which allows in principle the range $[0; \hat{\pi}_0]$ to be explored. Last, we also set $\forall i, \hat{\pi}_0^i = \pi_{max} >> \pi^i$.

3.2.2. Recursive Bayesian Estimation

Given our linear model Eq. (5), we wish to compute an optimal estimate of $\boldsymbol{\pi}$ by the conditional expectation,

$$\hat{\boldsymbol{\pi}}_k = \mathbb{E}[\boldsymbol{\pi}_k | \boldsymbol{\rho}_{0:k}], \tag{7}$$

where $\hat{\boldsymbol{\pi}}_k$ is the optimal estimate of $\boldsymbol{\pi}_k$ given the sequence of observations $\boldsymbol{\rho}_0$ to $\boldsymbol{\rho}_k$, which we denote $\boldsymbol{\rho}_{0:k}$. Note that, Eq. (7) requires one to compute the posterior probability density function (PDF) $p(\boldsymbol{\pi}_k | \boldsymbol{\rho}_{0:k})$,

$$p(\boldsymbol{\pi}_k | \boldsymbol{\rho}_{0:k}) = \frac{p(\boldsymbol{\pi}_k | \boldsymbol{\rho}_{0:k-1}) \cdot p(\boldsymbol{\rho}_k | \boldsymbol{\pi}_k)}{p(\boldsymbol{\rho}_k | \boldsymbol{\rho}_{0:k-1})}, \tag{8}$$

where,

$$p(\boldsymbol{\pi}_k | \boldsymbol{\rho}_{0:k-1}) = \int p(\boldsymbol{\pi}_k | \boldsymbol{\pi}_{k-1}) \cdot p(\boldsymbol{\pi}_{k-1} | \boldsymbol{\rho}_{0:k-1}) d\boldsymbol{\pi}_{k-1}, \tag{9}$$

is a recursive expression of the state at time k as a function of time $k-1$ and the most recent observations.

Typically, one distinguishes two phases during recursive Bayesian filtering: prediction and correction phase. In the prediction phase, we compute the a-priori state density (Eq. 9) using the transition function (Eq. 5). In the correction phase, a new observation vector is available to compute the likelihood $p(\boldsymbol{\rho}_k | \tilde{\boldsymbol{\pi}}_k)$ and normalization constant, whereby allowing the a-posteriori state estimate to be computed using Eq. (8).

Modeling the states as a multi-variate gaussian random variable with additive gaussian noise greatly simplifies this computation, especially when assuming that the transition and observation models are linear. In particular, the latter assumptions typically allow for Kalman Filter type solutions (Kalman, 1960). However, the present scenario imposes an inequality constraint on the states so as to make them interpretable as probabilities, a requirement that standard Kalman Filters does not allow for. Instead, Gupta and Hauser (2007) suggests applying an intermediate step in which the a-priori state estimates are projected on the constraint surface. This approach, despite being effective, requires the solving a quadratic program at each iteration.

Instead, we use the simpler approach of Kandepu et al. (2008), which relies on the Unscented Kalman Filter (UKF) approach (Wan and Van Der Merwe, 2000). In contrast with standard Kalman Filters, which propagate the means and covariances of states through the (linear) system, UKF samples a set of carefully chosen points from the state distribution, called *sigma-points*, that allow to accurately approximate the true statistics of the posterior. Our inequality constraints are then directly applied to the sigma-points.

As illustrated in Fig. 2, our self-supervised approach iteratively decreases its estimates of π^i (red line) as a function of observations ρ_k^i (orange line). Given the defined control inputs u_k which induce a downward acceleration, π^i are reduced further than they should be (see epoch 80 in Fig. 2). For this reason, we introduce a strategy to halt this computation at an appropriate moment in the next section.

3.2.3. Stopping condition

We introduce two stopping conditions that we use simultaneously to ensure that our methods stops at an appropriate value for $\hat{\pi}$.

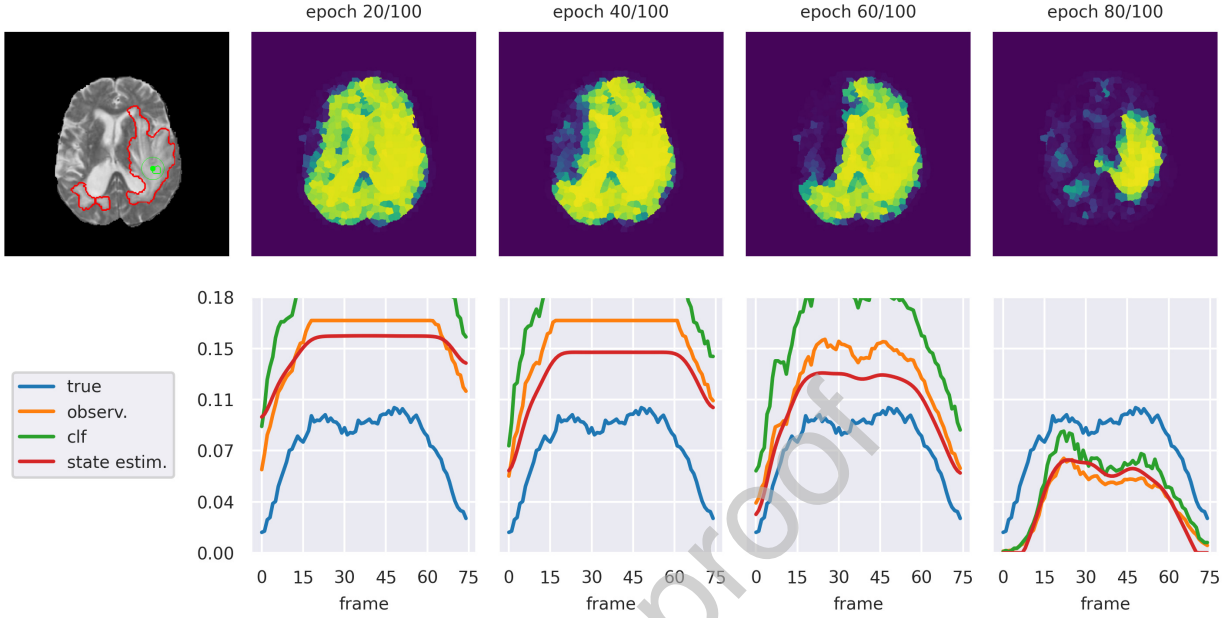
Specifically, first we denote $\tilde{\boldsymbol{X}}_n = \{x \in \boldsymbol{X} | f_\theta(x) < 0.5\}$, and $\tilde{\boldsymbol{X}}_p = \{x \in \boldsymbol{X} | f_\theta(x) \geq 0.5\}$ as the set of "pseudo-negative" and "pseudo-positive" samples, respectively. As a first criteria, we use the variance of the predictions of $\tilde{\boldsymbol{X}}_n$, written $Var[f_\theta(\tilde{\boldsymbol{X}}_n)]$, which measures the confidence of our model on the negative samples (*i.e.,* background regions of the image). Second, we also impose that our predictions are such that the frequency of positives is below our upper-bound on all frames. By combining these, our criteria (1) imposes a maximum on the frequencies of pseudo-positives, (*i.e.,* $\frac{|\tilde{\boldsymbol{X}}_p^i|}{N_i} < \hat{\pi}_0 \quad \forall i$) and (2) imposes a maximum variance level to pseudo-negatives (*i.e.,* $Var[f_\theta(\tilde{\boldsymbol{X}}_n)] < \tau$).

In practice, we want the above conditions to be verified for several iterations so as to guarantee stability. We therefore impose that both conditions are satisfied for T_s iterations. In Fig. 2, we illustrate the behaviour of our stopping conditions by showing the predicted probabilities and their corresponding class-priors on a given sequence.

3.3. Spatio-Temporal Regularization

While our **SSnnPU** method leverages all images and pointwise annotations to train and segment the data volume in question, the output of our method does not explicitly leverage the

Fig. 2. Visualization of our SSnnPU method on an example MRI volume. (Top row): Single slice from volume with groundtruth segmentation highlighted (red) and a point-wise annotation (green), output prediction at different epochs. (Bottom row): Class priors at corresponding epochs over 75 slices of the volume. The different curves correspond to: the true priors (blue), observations ρ_k (orange), proportion of pseudo-positives (green), and current estimated priors $\hat{\pi}_k$ (red). The stopping condition is triggered at epoch 71. Figure best seen in color.



spatio-temporal relations within the data cube. That is, every sample is treated and predicted independently, and only implicitly related through f_θ. In order to coherently regularize over the different frames and locations, we use an existing graph based framework as a post-processing step.

To this end, we make use of the multi-object tracking framework (**KSPTrack**) introduced in Lejeune et al. (2018) to refine the output of the **SSnnPU** method. In short, **KSPTrack** represents the data volume with superpixels and builds a network graph over these to optimize a set of spatio-temporal paths that jointly correspond to the object segmentation throughout the data volume. This is solved by casting the problem as network-flow optimization, whereby costs are assigned to input/output nodes, visiting and transition edges within and across frames, and where 2D annotations are used to define source nodes that allow to push flow within the network.

In practice, we use the same orginal **KSPTrack** setup as in Lejeune et al. (2018) with the exception of using the output of **SSnnPU**, $f_\theta(x_i)$ to compute the cost of selecting superpixel x_i as part of the object by,

$$C_{fg}(i) = -\log \frac{f_\theta(x_i)}{1 - f_\theta(x_i)}, \tag{10}$$

where $C_{fg}(i)$ is the cost of including superpixed x_i as part of the object. By construction, this relation therefore imposes a negative cost when $f_\theta(x) > 0.5$, and a non-negative cost otherwise.

The final output of the **KSPTrack** method yields a binary image for each of the frames in the data volume. For the remainder of this paper, we will refer to the combined use of **SSnnPU** and **KSPTrack** as **SSnnPU+KSPTrack**.

3.4. Training details, hyper-parameters, and implementation

We now specify technical details of our implementation and training procedure. **SSnnPU** is implemented in Python using the PyTorch library[1], while we use a publicly available C++ implementation of the K-shortest paths algorithm for the spatio-temporal regularization step[2].

3.4.1. SSnnPU

f_θ is implemented as a Convolutional Neural Network based on the U-Net architecture proposed in Ibtehaz and Rahman (2020) for all experiments. It uses "Inception-like" blocks in place of simple convolutional layers to improve robustness to scale variations. Skip connections are replaced by a serie of 3×3 convolutional layers with residual connections. Batch normalization layers are added after each convolutional layer.

To train **SSnnPU**, we proceed with a three phase process:

1. To increase the robustness of early observations, we train f for 50 epochs with Alg. 1 and a learning rate set to 10^{-4}. With the last layer of our decoder being a sigmoid function, we set the bias of the preceding convolutional layer to $-\log\frac{1-\pi_{init}}{\pi_{init}}$, with $\pi_{init} = 0.01$, as advised in Lin et al. (2017). All others parameters are initialized using He's method He et al. (2015).
2. We then optimize the model and class-prior estimates for a maximum 100 epochs as described in Alg. 2 with a learning rate set to 10^{-5}.

[1] https://github.com/lejeunel/ssnnpu_ksptrack
[2] https://github.com/lejeunel/pyksp

3. We then train using frame-wise priors given by the previous phase for an additional 100 epochs with a learning rate of 10^{-5}.

We use the Adam optimizer with weight decay 0.01 for all training phases. Data augmentation is performed using a random combination of additive gaussian noise, bilateral blur and gamma contrasting.

3.4.2. Recursive Bayesian Estimation

For the process, transition, and initial covariance matrices, we use diagonal matrices $Q = \sigma_Q \mathbb{I}$, $R = \sigma_R \mathbb{I}$, and $S = \sigma_S \mathbb{I}$, where $\mathbb{I}$ is the identity matrix. As the observations ρ_k^i are often very noisy, we set $\gamma = 2$ and the observation variance much larger than the process variance $\sigma_Q = 10$, $\sigma_R = 0.05$ and $\sigma_S = 0.03$. The parameters of the control input are set proportionally to $\hat{\pi}_0$ with $u_0 = 0.02\hat{\pi}_0$, and $u_T = 0.4\hat{\pi}_0$. The window length of the frame-wise smoothing filter is set proportionally to the number of frames: $L = 0.05N$. The time-period of our stopping condition is set to $T_s = 10$ and the threshold on the variance is $\tau = 0.007$.

3.4.3. KSPTrack parameters

All sequences are pre-segmented into $\sim$ 1200 superpixels and the output of f is averaged over all pixels in a superpixel. Each point-wise annotation defines a circle of radius $R = 0.05 \cdot \max\{W, H\}$ centered on the 2D location, where W and H are the width and height of frames, respectively. The input cost at given superpixel is set to 0 when its centroid is contained within that circle, and ∞ otherwise. The transitions costs are set to 0 when superpixels overlap and ∞ otherwise. In order to reduce the number of edges and alleviate the computational requirements, we also prune visiting edges when their corresponding object probability falls below 0.4. We perform a single round of **KSPTrack** as augmenting the set of positives and re-training the object model after each round (as in Lejeune et al. (2018)) did not prove beneficial.

4. Experiments

In the following section, we outline the experiments performed to characterize the behavior of our method. First, we compare our method with existing baselines for segmentation purposes. We then perform an ablation study to demonstrate which aspect of our method provides what quantitative benefits, as well as the impact of the class-prior upper-bound initialization. Last, we show how our stopping condition performs in establishing useful class priors.

4.1. Datasets

To validate our method, we evaluate it on the publicly available dataset used in Lejeune et al. (2018)[3]. It consists of a variety of video and volumes of different modalities with 2D annotation points for different objects of interest, as well as the associated groundtruth segmentations. Specifically, it includes four different subsets of data:

- **Brain:** Four 3D T2-weighted MRI scans chosen at random from the publicly available BRATS challenge dataset (Menze, 2014), where tumors are the object of interest.
- **Tweezer:** Four sequences from the training set of the publicly dataset MICCAI Endoscopic Vision challenge: Robotic Instruments segmentation. The surgical instrument is the object to segment in these sequences.
- **Slitlamp:** Four slit-lamp video recordings of human retinas, where the optic disk is to be segmented.
- **Cochlea:** Four volumes of 3D CT scans of the inner ear, where the cochlea must be annotated. This object is the most challenging object to segment due to its challenging geometry (*i.e.,* non-concave shape).

4.2. Baselines and experimental setup

Using the datasets mentioned above, we evaluate the proposed methods (**SSnnPU** and **SSnnPU+KSPTrack**) quantitatively and qualitatively. Additionally, we compare these to existing baseline methods that perform the same tasks. These include:

- **KSPTrack**: Multi-object tracking method described in Lejeune et al. (2018). As in the original work, the object model consists of a decision tree bagger adapted to the PU setting, while features are taken from a CNN configured as an autoencoder.
- **EEL**: An expected exponential loss within a boosting framework for robust classication in a PU learning setting (Lejeune et al., 2017).
- **Gaze2Segment**: A learned saliency-map based detection regularized with graphcut (Khosravan et al., 2017).
- **DL-prior**: Point location supervision is used to train a CNN with strong object priors (Bearman et al., 2016).
- **AlterEstPU**: An alternating training and class-prior estimation method inspired by the expectation-maximization algorithm that leverages the non-negative Positive/Unlabeled risk estimator of Kiryo et al. (2017) as described in Kato et al. (2018). This uses the same model, training scheme and parameters as **SSnnPU** (Sec. 3.4), with the exception that the second phase is replaced by the class-prior update scheme of Kato et al. (2018). Note that in contrast to **SSnnPU**, a single value for the class-prior is estimated. We tune the parameters of the update scheme for best performance on the tested sequences.

Methods **ELL** and **KSPTrack** have been specifically designed for the proposed evaluation datasets (sec. 4.1), and their parameters have been optimized for best performance. Methods **Gaze2Segment** and **DL-prior** have been implemented and optimized to give their best performances.

[3]Datasets, manual ground truth annotations, and point-wise annotations used in this paper are available at `https://doi.org/10.5281/zenodo.5007788`

For our method, we set the initial class prior upper-bound, π_{max}, by computing the frequencies of positives π^i for all frames from the groundtruth and set $\hat{\pi}_0 = 1.4 \cdot \max_i\{\pi^i\}$. While this may appear to be using the groundtruth to set the parameters of our method, it allows us to calibrate π for comparison reasons. Indeed, a 40% factor over the frame that has the largest object surface object can represent the entire image depending on the dataset. To show that our method is not inherently sensitive to this value, we perform an analysis of sensitivity in Sec. 4.4.

Following a transductive learning setup, the input of each of evaluated methods is a single data volume and their associated 2D point annotations, and yields a complete segmentation for the inputted data volume.

4.3. Segmentation performance

In Table. 1, we show the F1 scores of each method, averaged over each sequence for each data type. Given that some methods require superpixels, we also show the maximum performance a segmentation method would have if every single superpixel was correctly labeled. In practice we denote positive superpixels as those with more than half of the pixels are in the groundtruth. We then compare the latter segmentation with the pixel-wise manual ground truth annotation and denote the result **Max. SP**.

From these results, we note that both **SSnnPU** and **SSnnPU+KSPTrack** perform well on average. Most notably, **SSnnPU+KSPTrack** outperformes all other approaches including **SSnnPU**. This is coherent as the spatio-temporal regularization provides an efficient method to remove false positives generated by **SSnnPU**. On the Tweezer sequences, the gain in performances are striking, with an increase in 14% on average over the previous state-of-the-art, and closely approaches a perfect labeling according to **Max. SP**.

When comparing **SSnnPU** and **SSnnPU+KSPTrack**, we note that the spatio-temporal regularization provided by the KSP framework is particularly effective in the Cochlea cases (*i.e.,* plus 18%). This is coherent as the geometry of the cochlea is largely made of a rings and where visual appearance plays a somewhat lesser role in identifying the complete structure. This latter point also explains the fairly poor performance of **SSnnPU** but much improved one by **SSnnPU+KSPTrack** when compared to **KSPTrack**.

Last, comparing **SSnnPU** and **AlterEstPU**, we observe important limitations of the latter method, which, aside from the fact that it only estimates a single value for the class-prior over the whole sequence, tends to be very sensitive to noise and often fails to converge, thereby showing inferior performance and higher variance.

In Fig. 3, we show qualitative results of different methods and provide complete video results of **SSnnPU+KSPTrack** as supplementary material.

4.4. Ablation study

To provide a better understanding as to what aspects of our method provide improvements, we perform an ablation study. Specifically, we evaluate the following variants of our methods:

- **SSnnPU**: Non-negative positive-unlabeled with self-supervised class-prior estimation, as in Sec. 3.1 and 3.2.
- **SSnnPU+KSPTrack**: Combines the **SSnnPU** method and the **KSPTrack** methods described in Sec. 3.3.
- **nnPUTrue+KSPTrack**: Same as **SSnnPU+KSPTrack**, except that the foreground model is directly trained using the true class-priors (*i.e.,* taken from the groundtruth) following Alg. 1.
- **nnPUConst+KSPTrack**: Same as **nnPUTrue+KSPTrack**, except that we use a sequence-wise constant class prior given by the mean groundtruth prior over all frames (as in Kiryo et al. (2017)).

For both **nnPUTrue+KSPTrack** and **nnPUConst+KSPTrack**, we train the models f for 150 epochs. The learning rate in both cases are set to 10^{-4} and reduced to 10^{-5} after 50 epochs.

In addition, to assess how **SSnnPU+KSPTrack** performs as a function of the selected π_{max}, we evaluate the method using $\hat{\pi}_0 = \eta \max_i \pi^i$, with $\eta \in \{1.2, 1.4, 1.6, 1.8\}$. Similarly, to assess the relevance of the self-supervised estimation of priors, we perform segmentation using method **nnPUConst+KSPTrack** with $\eta \in \{0.8, 1.0, 1.2, 1.4\}$.

We report F1, precision and recall scores for each aforementioned method in Table. 2. First, we observe that our self-supervised estimation is relatively robust to variations in $\hat{\pi}_0$. In particular, Tweezer shows variations in F1 scores of at most 1% while η ranges between 1.2 and 1.6. Similarly, the performances fluctuate only marginally for Cochlea, Slitlamp and Brain sequences with at most 5% changes. These fluctuations still lead to improved performances over state-of-the-art methods.

In contrast, sensitivity to initial class priors is much stronger for **nnPUConst+KSPTrack**, which yields variations between 1% and 15% depending on the sequence type. The relevance of the frame-wise prior estimation can also be assessed by comparing, for each sequence type, the maximum F1 scores reached by **SSnnPU+KSPTrack** and **nnPUConst+KSPTrack** for all tested values of η. **SSnnPU+KSPTrack** brings an improvement over **nnPUConst+KSPTrack** of 2%, 1%, 6% and 0% for the Tweezer, Brain, Cochlea and Slitlamp sequence, respectively.

Last, comparing **SSnnPU+KSPTrack** to **nnPUTrue+KSPTrack**, we note that the proposed self-supervised prior estimation framework brings comparable performances on all types. That is the **SSnnPU** component of our methods appears to provide class priors on par with the true class-priors. Surprisingly, in the Tweezer case, some values of η yield even better performances than if the true class prior values were to be used.

4.5. Analysis of Convergence

Last, we analyze the behavior of our proposed stopping conditions in Alg. 2. In particular, Fig. 4 illustrates the convergence of the class prior estimation for each type of sequence. In these cases, all sequences are trained using $\hat{\pi}_0 = 1.4 \cdot \max_i\{\pi^i\}$.

Table 1. Quantitative results on all datasets. We report the F1 scores and standard deviations. In column "All", we show the average F1 score on all sequences. In column Δ, we show the absolute difference with respect to the maximum achievable score given the superpixel over-segmentation (Max. SP).

Types Methods	Tweezer	Cochlea	Slitlamp	Brain	All	Δ
Max. SP	0.92 ± 0.02	0.92 ± 0.01	0.92 ± 0.02	0.95 ± 0.01	0.93	-
SSnnPU+KSPTrack	**0.91** ± 0.03	**0.75** ± 0.05	**0.84** ± 0.05	**0.80** ± 0.09	**0.82**	**−0.10**
SSnnPU	0.87 ± 0.02	0.53 ± 0.10	0.78 ± 0.10	0.75 ± 0.13	0.73	−0.19
KSPTrack	0.77 ± 0.08	0.66 ± 0.02	0.77 ± 0.08	0.74 ± 0.08	0.74	−0.19
AlterEstPU	0.74 ± 0.12	0.39 ± 0.11	0.65 ± 0.11	0.53 ± 0.24	0.58	−0.35
EEL	0.60 ± 0.16	0.12 ± 0.05	0.59 ± 0.08	0.52 ± 0.14	0.46	−0.47
Gaze2Segment	0.18 ± 0.00	0.07 ± 0.02	0.02 ± 0.00	0.07 ± 0.02	0.08	−0.84
DL-prior	0.72 ± 0.06	0.30 ± 0.04	0.51 ± 0.11	0.56 ± 0.08	0.52	−0.40

As we aim to estimate the true class-prior, we plot the Mean Absolute Error between the estimated priors and the true priors (top panels). The proposed stopping condition, which leverages the variance of pseudo-negatives, is plotted in the bottom panel. We observe that the proposed stopping condition triggers reasonably close to the optima in most cases. In cases where the stopping condition is triggered earlier or later, we note that the difference in mean absolute error is fairly small, whereby implying that the impact in the early/late trigger is rather small.

5. Conclusions

The present work contributes to the challenging problem of segmenting medical sequences of various modalities using point-wise annotations and without knowing in advance what is to be segmented. As such, it has important implications in the ability to quickly produce groundtruth annotations or learn from a single example.

By formulating our problem as a positive/unlabeled prediction task, we demonstrated the relevance of the non-negative unbiased risk estimator as a loss function of a Deep Convolutional Neural Network. Our novel contribution of a self-supervised framework based on recursive bayesian filtering, to estimate the class priors, a hyper-parameter that plays an important role in the segmentation accuracy, was demonstrated to bring important performance gains over state-of-the-art methods, particularly when also used in combination with a spatio-temporal regularlization scheme. From the annotator's point of view, the burden is marginally increased in what the method requires, asides from 2D locations and an upper-bound on the class-prior. While our approach does not perform flawlessly in challenging cases, we show that the performance is stable and resilient to miss-specified class prior upper-bounds. Last, we show that our stopping conditions are adept at yielding favorable estimates as well.

As future works, we aim at investigating the addition of negative samples into our foreground model, by leveraging the positive, unlabeled, and biased negative risk estimator of Hsieh et al. (2019). Similarly, given that our regularization is outside of the learning framework, we wish to also explore how this could be integrated to increase performance. Last, there are interesting potentials in using recent graph-based PU learning approaches Akujuobi et al. (2020) to see how additional data can be sequentially added and annotated. In this way, trained models per sequence could be aggregated as more data is available.

Acknowledgments

This work was supported in part by the Swiss National Science Foundation Grant 200021 162347 and the University of Bern.

References

Akujuobi, U., Chen, J., Elhoseiny, M., Spranger, M., Zhang, X., 2020. Temporal positive-unlabeled learning for biomedical hypothesis generation via risk estimation, in: Advances in Neural Information Processing Systems 33.

Bearman, A., Russakovsky, O., Ferrari, V., Fei-Fei, L., 2016. What's the Point: Semantic Segmentation with Point Supervision. European Conference on Computer Vision .

Bekker, J., Davis, J., 2018. Estimating the class prior in positive and unlabeled data through decision tree induction, in: Proceedings of the 32th AAAI conference on artificial intelligence, pp. 2712–2719.

Bekker, J., Davis, J., 2020. Learning from positive and unlabeled data: a survey. Mach. Learn. 109, 719–760.

Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. 23, 1222–1239.

Chen, L., Bentley, P., Mori, K., Misawa, K., Fujiwara, M., Rueckert, D., 2019. Self-supervised learning for medical image analysis using image context restoration. Medical image analysis 58, 101539.

Christoffel, M., Niu, G., Sugiyama, M., 2016. Class-prior estimation for learning from positive and unlabeled data, in: Asian Conference on Machine Learning, pp. 221–236.

Du Plessis, M.C., Sugiyama, M., 2014. Class prior estimation from positive and unlabeled data. IEICE TRANSACTIONS on Information and Systems 97, 1358–1362.

Elkan, C., Noto, K., 2008. Learning classifiers from only positive and unlabeled data, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 213–220.

Ferreira, P.M., Mendonça, T., Rozeira, J., Rocha, P., 2012. An annotation tool for dermoscopic image segmentation, in: Proceedings of the 1st International Workshop on Visual Interfaces for Ground Truth Collection in Computer Vision Applications, pp. 1–6.

Gupta, N., Hauser, R., 2007. Kalman filtering with equality and inequality state constraints. arXiv preprint arXiv:0709.2791 .

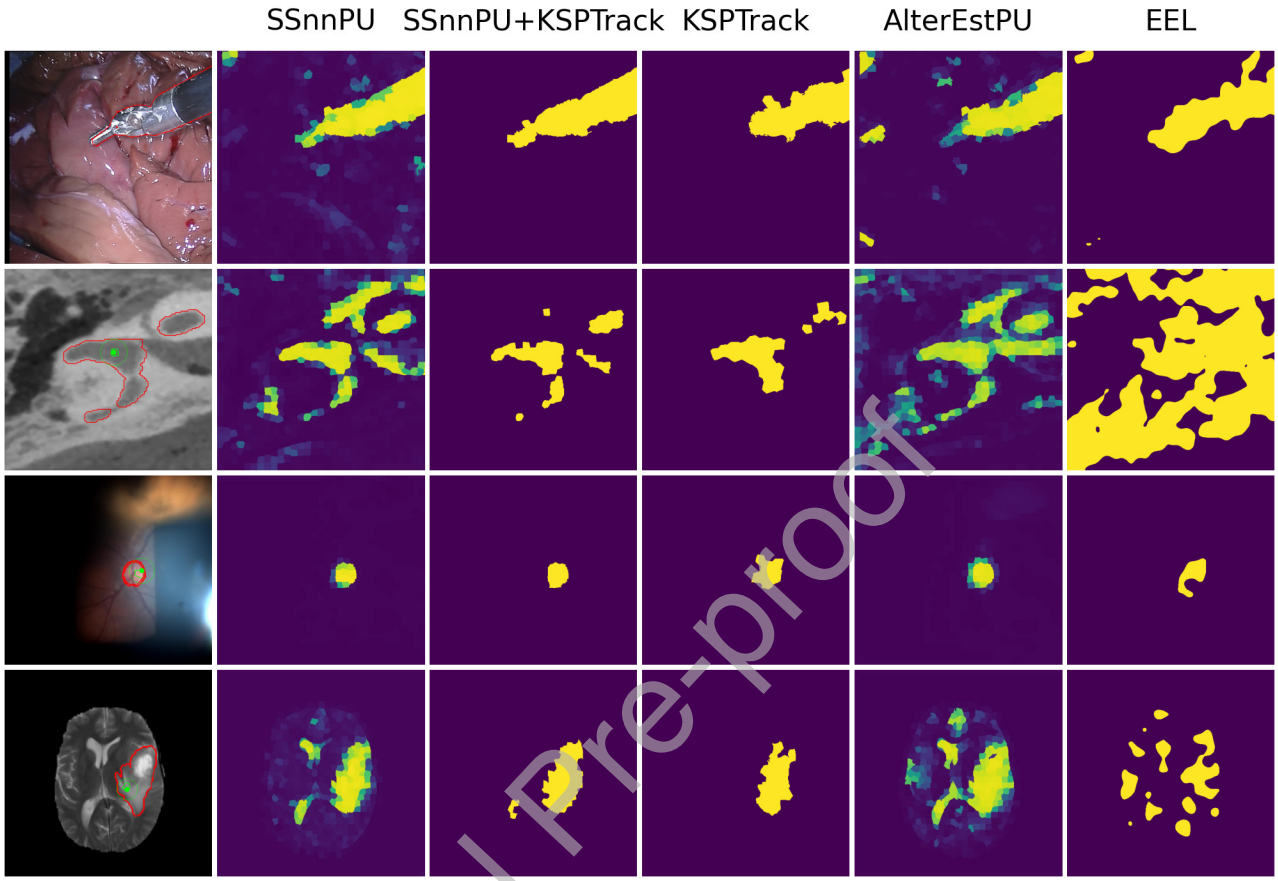
He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR abs/1502.01852.

Heim, E., Roß, T., Seitel, A., März, K., Stieltjes, B., Eisenmann, M., Lebert, J., Metzger, J., Sommer, G., Sauter, A.W., et al., 2018. Large-scale medical image annotation with crowd-powered algorithms. Journal of Medical Imaging 5, 034002.

Table 2. Quantitative results on all datasets for different prior levels. We report the F1 score, precision (PR), recall(RC) and standard deviations.

Types	Methods	η	F1	PR	RC
Tweezer	nnPUconst+KSPTrack	0.8	0.89 ± 0.03	0.91 ± 0.05	0.88 ± 0.03
		1.0	0.90 ± 0.04	0.86 ± 0.06	0.93 ± 0.02
		1.2	0.90 ± 0.03	0.84 ± 0.05	0.96 ± 0.01
		1.4	0.89 ± 0.03	0.83 ± 0.05	0.97 ± 0.00
	SSnnPU+KSPTrack	1.2	0.92 ± 0.01	0.93 ± 0.02	0.92 ± 0.03
		1.4	0.91 ± 0.03	0.91 ± 0.03	0.91 ± 0.05
		1.6	0.91 ± 0.03	0.88 ± 0.03	0.94 ± 0.03
		1.8	0.90 ± 0.04	0.88 ± 0.04	0.92 ± 0.04
	nnPUtrue+KSPTrack	-	0.90 ± 0.03	0.89 ± 0.03	0.92 ± 0.04
Cochlea	nnPUconst+KSPTrack	0.8	0.59 ± 0.15	0.88 ± 0.09	0.45 ± 0.15
		1.0	0.67 ± 0.07	0.87 ± 0.14	0.55 ± 0.06
		1.2	0.71 ± 0.10	0.91 ± 0.10	0.59 ± 0.12
		1.4	0.68 ± 0.10	0.77 ± 0.18	0.62 ± 0.11
	SSnnPU+KSPTrack	1.2	0.73 ± 0.08	0.85 ± 0.16	0.65 ± 0.03
		1.4	0.75 ± 0.05	0.88 ± 0.08	0.66 ± 0.04
		1.6	0.72 ± 0.07	0.80 ± 0.20	0.68 ± 0.07
		1.8	0.64 ± 0.14	0.80 ± 0.32	0.58 ± 0.07
	nnPUtrue+KSPTrack	-	0.76 ± 0.05	0.85 ± 0.08	0.69 ± 0.03
Slitlamp	nnPUconst+KSPTrack	0.8	0.71 ± 0.30	0.84 ± 0.08	0.72 ± 0.38
		1.0	0.84 ± 0.03	0.79 ± 0.05	0.91 ± 0.04
		1.2	0.78 ± 0.05	0.66 ± 0.07	0.95 ± 0.01
		1.4	0.66 ± 0.04	0.51 ± 0.05	0.95 ± 0.01
	SSnnPU+KSPTrack	1.2	0.72 ± 0.29	0.92 ± 0.05	0.67 ± 0.34
		1.4	0.84 ± 0.05	0.85 ± 0.08	0.84 ± 0.12
		1.6	0.80 ± 0.11	0.89 ± 0.06	0.75 ± 0.19
		1.8	0.84 ± 0.04	0.77 ± 0.06	0.92 ± 0.02
	nnPUtrue+KSPTrack	-	0.84 ± 0.03	0.79 ± 0.04	0.90 ± 0.02
Brain	nnPUconst+KSPTrack	0.8	0.79 ± 0.09	0.82 ± 0.18	0.78 ± 0.04
		1.0	0.78 ± 0.11	0.77 ± 0.12	0.79 ± 0.10
		1.2	0.72 ± 0.09	0.60 ± 0.10	0.92 ± 0.06
		1.4	0.73 ± 0.08	0.60 ± 0.09	0.93 ± 0.06
	SSnnPU+KSPTrack	1.2	0.79 ± 0.10	0.82 ± 0.14	0.76 ± 0.07
		1.4	0.80 ± 0.09	0.78 ± 0.14	0.84 ± 0.07
		1.6	0.77 ± 0.10	0.80 ± 0.12	0.75 ± 0.12
		1.8	0.76 ± 0.11	0.80 ± 0.11	0.73 ± 0.17
	nnPUtrue+KSPTrack	-	0.80 ± 0.09	0.73 ± 0.10	0.89 ± 0.07

Fig. 3. Qualitative results for each type. From left to right: Original image with groundtruth highlighted in red and 2D location in green, output of foreground prediction model (SSnnPU), SSnnPU combined with our spatio-temporal regularization scheme (SSnnPU+KSPTrack), best baseline (KSPTrack), second best baseline (AlterEstPU), and third best baseline (EEL).



Hsieh, Y.G., Niu, G., Sugiyama, M., 2019. Classification from positive, unlabeled and biased negative data, in: International Conference on Machine Learning, PMLR. pp. 2820–2829.

Ibtehaz, N., Rahman, M.S., 2020. Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation. Neural Networks 121, 74–87.

Jamaludin, A., Kadir, T., Zisserman, A., 2017. Self-supervised learning for spinal mris, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, pp. 294–302.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. Journal of Basic Engineering .

Kandepu, R., Imsland, L., Foss, B.A., 2008. Constrained state estimation using the unscented kalman filter, in: 2008 16th Mediterranean Conference on Control and Automation, IEEE. pp. 1453–1458.

Kato, M., Xu, L., Niu, G., Sugiyama, M., 2018. Alternate estimation of a classifier and the class-prior from positive and unlabeled data. CoRR abs/1809.05710.

Khosravan, N., Celik, H., Turkbey, B., Cheng, R., McCreedy, E., McAuliffe, M., Bednarova, S., Jones, E., Chen, X., Choyke, P., Wood, B., Bagci, U., 2017. Gaze2segment: A pilot study for integrating eye-tracking technology into medical image segmentation, in: Workshop on Medical Computer Vision, Internation Conference on Medical Image Computing and Computer Aided Intervention, pp. 94–104.

Kiryo, R., Niu, G., Du Plessis, M.C., Sugiyama, M., 2017. Positive-unlabeled learning with non-negative risk estimator, in: Advances in neural information processing systems, pp. 1675–1685.

Knbelreiter, P., Sormann, C., Shekhovtsov, A., Fraundorfer, F., Pock, T., 2020. Belief propagation reloaded: Learning bp-layers for labeling problems, in: Computer Vision and Pattern Recognition.

Konyushkova, K., Sznitman, R., Fua, P., 2015. Introducing geometry in active learning for image segmentation, in: ICCV, pp. 2974–2982.

Lee, W.S., Liu, B., 2003. Learning with positive and unlabeled examples using weighted logistic regression, in: ICML, pp. 448–455.

Lejeune, L., Christoudias, M., Sznitman, R., 2017. Expected exponential loss for gaze-based video and volume ground truth annotation, in: Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis, International Conference on Medical Image Computing and Computer Assisted Intervention, pp. 106–115.

Lejeune, L., Grossrieder, J., Sznitman, R., 2018. Iterative multi-path tracking for video and volume segmentation with sparse point supervision. Medical Image Analysis 50, 65 – 81.

Li, X., Liu, B., 2003. Learning to classify texts using positive and unlabeled data, in: IJCAI, pp. 587–592.

Li, X.L., Liu, B., 2005. Learning from positive and unlabeled examples with different data distributions, in: European conference on machine learning, Springer. pp. 218–229.

Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P., 2017. Focal loss for dense object detection. CoRR abs/1708.02002.

Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S., 2003. Building text classifiers using positive and unlabeled examples, in: Third IEEE International Conference on Data Mining, IEEE. pp. 179–186.

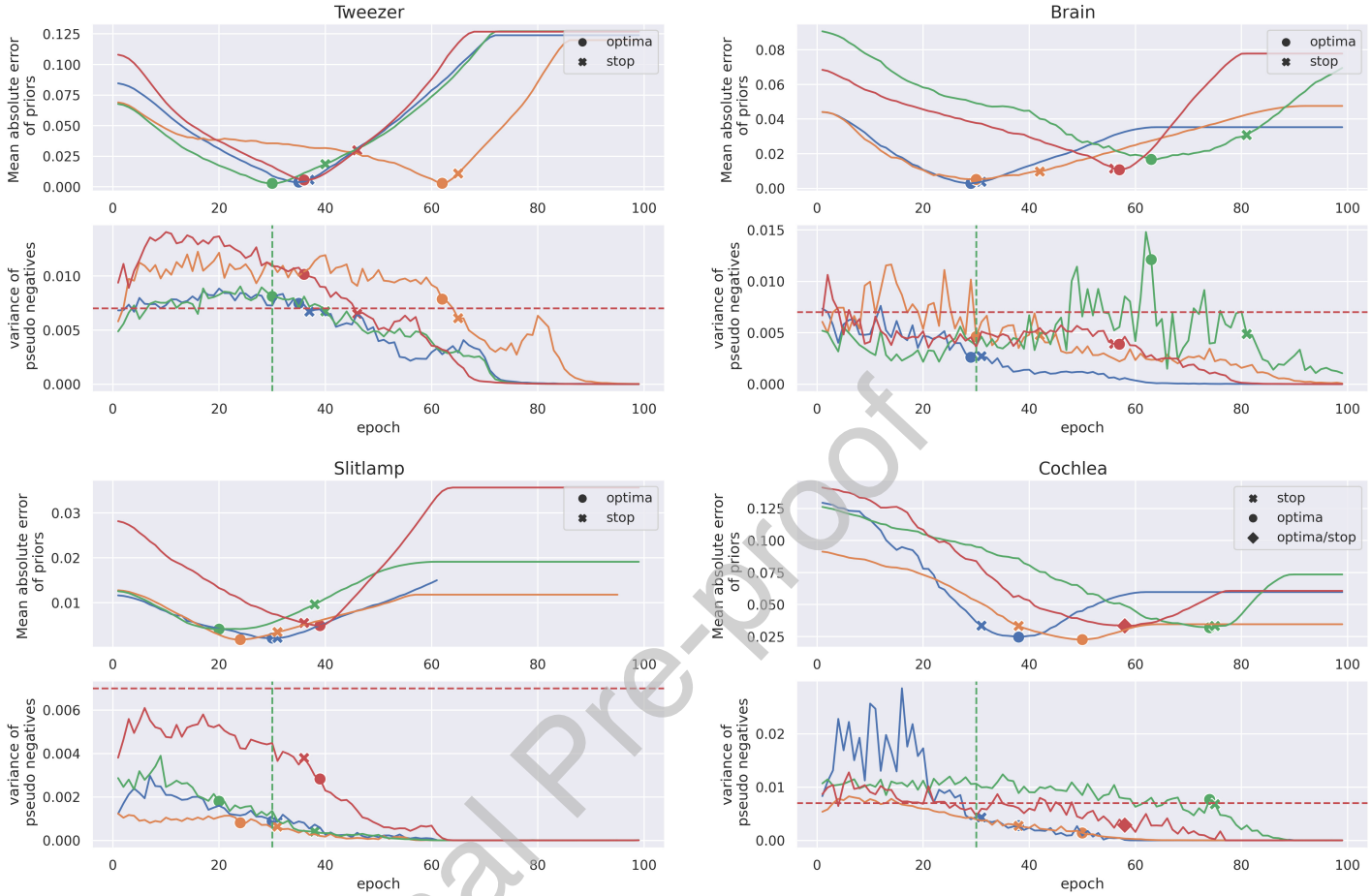
Menze, B.e., 2014. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). IEEE Transactions on Medical Imaging 34, 1993–2024.

Plessis, M.D., Niu, G., Sugiyama, M., 2015. Convex formulation for learning from positive and unlabeled data, in: International Conference on Machine Learning, pp. 1386 – 1394.

Salvador, A., Carlier, A., Giro-i Nieto, X., Marques, O., Charvillat, V., 2013. Crowdsourced object segmentation with a game, in: Proceedings of the 2nd ACM international workshop on Crowdsourcing for multimedia, pp. 15–20.

Scott, C., Blanchard, G., 2009. Novelty detection: Unlabeled data definitely

Fig. 4. Visualization of stopping conditions for SSnnPU method. For each tested image type, we show: (top) Mean Absolute Error (MAE) between the estimated and the true prior, (bottom) Variance of the pseudo-negatives. We also show here threshold (in dashed-red), and the minimum number of epochs (dashed-green). Both the optimal (circle) and the stopping conditions-based class-priors (cross) are shown on each of the sequences of each type (one line per sequence).



help, in: Artificial intelligence and statistics, pp. 464–471.

Sener, O., Savarese, S., 2018. Active learning for convolutional neural networks: A core-set approach. arXiv:1708.00489.

Vilariño, F., Lacey, G., Zhou, J., Mulcahy, H., Patchett, S., 2007. Automatic labeling of colonoscopy video for cancer detection, in: Pattern Recognition and Image Analysis: Iberian Conference, pp. 290–297.

Wan, E.A., Van Der Merwe, R., 2000. The unscented kalman filter for nonlinear estimation, in: Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373), Ieee. pp. 153–158.

Author Statement

Laurent Lejeune:Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization

Raphael Sznitman: Conceptualization, Methodology, Formal analysis, Investigation, Resources, Writing - Original Draft, Writing - Review & Editing, Supervision, Project administration, Funding acquisition

Declaration of Interest

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.