Universidad Zaragoza
1542

Master's Thesis

Leveraging foundation models to improve weakly supervised segmentation models in wildlife monitoring applications

Author

César Borja Moreno

Supervisors

Ana Cristina Murillo Arnal

Carlos Plou Izquierdo

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2025

Abstract

Semantic segmentation is a widely studied visual recognition task that focus on assigning a semantic label to each pixel in an image, offering a detailed understanding of the scene. However, training semantic segmentation models typically requires large amounts of high-quality pixel-level annotations. These annotations are often limited in many specific fields, as they require a significant human effort. Wildlife monitoring, and specially underwater imagery, is a clear example of a very relevant domain where such detailed annotations are scarce. This lack of pixel-level annotations and the huge human effort required to produce them, motivates the need to develop automatic tools that ease the labelling process required to train a semantic segmentation model for such a specific domain.

In this work, we propose to leverage powerful foundation models to develop weak supervision strategies that generate dense and detailed labels from limited annotations. This approach could significantly reduce the time spent on manual labelling, making ecological research more efficient and helping researchers analyze into the health and dynamics of wildlife environments. Specifically, we explore label augmentation focusing on the next challenge: generate a "dense" semantic segmentation of an underwater image from a set of sparse point-level labels provided by an expert. Our approach is built upon SAM2 segmentation and DINOv2 features extraction capabilities. It starts with the propagation of all sparse point-labels across the image which is followed by a posterior refinement of the propagated segmentation by predicting labels for the remaining unlabeled pixels. As result, we generate a dense semantic segmentation from minimal annotations.

The experiments demonstrate that our approach outperforms current state-of-the-art superpixel based method in terms of label augmentation quality. This improvement is particularly highlighted when we start from a extremely low number of point-labels ($\sim 0.01\%$ of image pixels) and when we qualitatively compare the mask shapes. Furthermore, we validate our approach, training a semantic segmentation model like SegFormer using only our augmented labels as supervision for the model. The results show that our SegFormer training strategy achieves competitive performance than when we trained it with dense ground truth labels.

Acknowledgements

I am truly grateful to my first supervisor Ana, who has guided me not only throughout this thesis but also during my bachelor's thesis. Her mentorship has been instrumental in my academic journey, and I look forward to continuing to learn from her during my PhD.

I would also like to extend my deepest thanks to Carlos, my other supervisor, who was always by my side in the lab. His constant support, hands-on guidance, and advice have been invaluable during the development of this work.

I would also like to express my sincere gratitude to The Byrnes Lab team from the UMass Boston, whose collaboration marked the beginning of this project. They provided valuable imagery and offered valuable feedback throughout the process. I am especially thankful for their hospitality in inviting me to their lab, where I had the opportunity to learn from their expertise firsthand.

Lastly, I would like to mention my colleagues in the lab who have welcomed me and made me feel like a part of the team from day one and make the daily routine fulfilling. In particular, I would like to give an special thanks to my classmate Nerea Gallego, who has shared this journey with me since our first year at university, being a constant source of support.

Contents

Chapter 1

Introduction

1.1 Motivation

Understanding ecosystems, biodiversity, and natural processes is essential for environmental research and conservation. Wildlife monitoring is a fundamental practice in this field, allowing researchers to gather critical data on the health and dynamics of ecosystems. However, there are some environments in which wildlife monitoring is quite challenging (Figure 1.1). In this context, the integration of autonomous systems, such as Autonomous Underwater Vehicles (AUVs) and drones, and artificial intelligence (AI) is revolutionizing these efforts, allowing efficient and automated data collection in remote and challenging environments [1, 2, 3].

Artificial visual recognition, a subfield of AI, aims to enable computers to understand and interpret visual information (images or videos) as humans do. Within this domain, **semantic segmentation** is a highly-explored task that involves assigning a semantic label to each pixel in an image, providing a detailed understanding of the scene (Figure 1.2) [4]. To accurately address this task, semantic segmentation models require an extensive training with a significant amount of high-quality pixel-level annotated data. Unfortunately, in many scientific fields, such as marine biology, this kind of detailed annotations are often scarce due to specific-domain interests and the significant human effort required for annotations. Much of the available annotations are scarce labels and rely on sparse point-level labels, bounding boxes, text prompts, or other limited forms of annotation. In such specific fields in which ground truth data is scarce, other strategies must be explored to develop powerful segmentation models.

Foundation models are large-scale models trained on massive and diverse datasets, acquir-

(a) Underwater coral reef. (b) Dense seaweed bed. (c) African savanna.

Figure 1.1: Different wildlife monitoring environments in which wildlife monitoring is challenging.

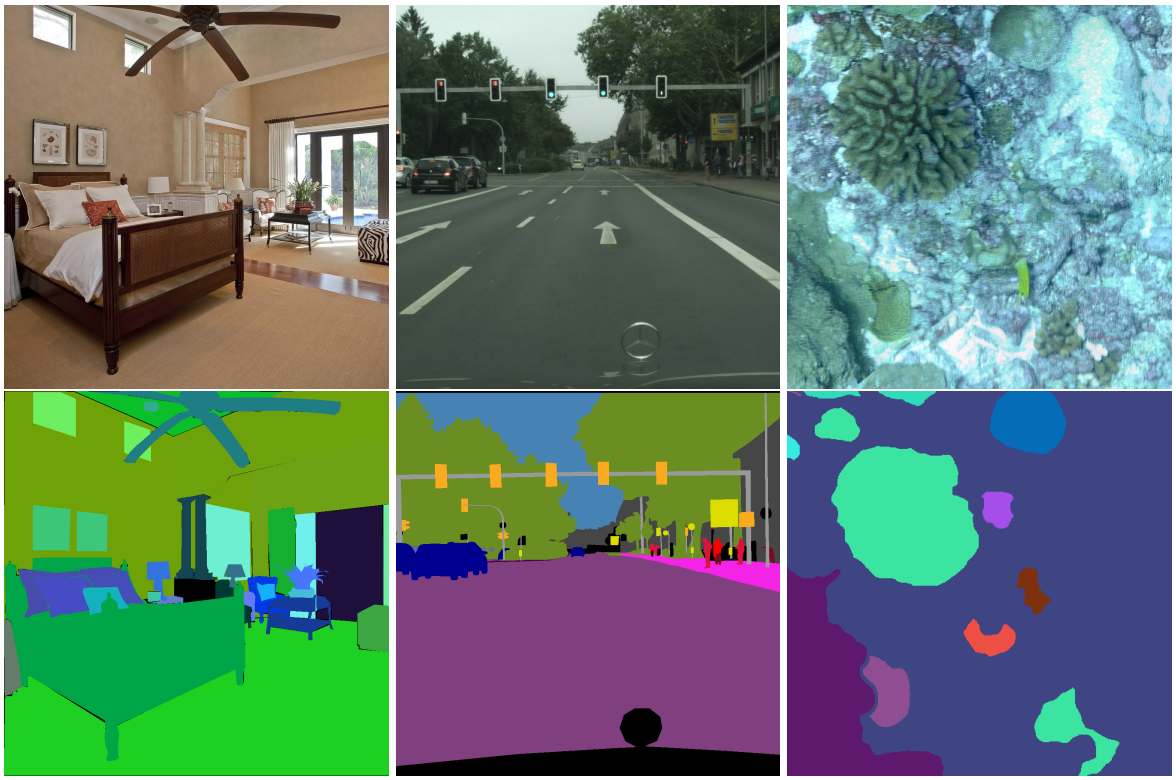


Figure 1.2: Semantic segmentation examples in a variety of scenarios.

ing a world knowledge that enables them to achieve a remarkable performance across a wide range of tasks. For example, in visual recognition, models such as DINOv2 [5], CLIP [6], and SAM (Segment Anything Model) [7] have demonstrated remarkable capabilities across various visual recognition tasks, including semantic segmentation. By exploiting their capabilities, we can build a **weak supervision** strategy that addresses the challenge to generate dense and detailed labels from scarce ground truth labels. This advance has the potential to significantly reduce the reliance on labor-intensive annotation, facilitating more efficient monitoring tasks and ecological applications, and ultimately improving the accuracy and scope of biodiversity studies. These motivations lead us directly to the main objective described in the next section.

1.2 Goal and tasks

The overall goal of this master's thesis is to investigate and develop weakly supervised segmentation strategies leveraging the power of visual recognition foundation models in the context of wildlife monitoring. Specifically, we will develop a system capable of generating "dense" labels from sparse point-level labels. The work will focus on underwater imagery from different wildlife scenarios. To achieve this goal, the following tasks were accomplished throughout the project (its distribution along the months may be observed in Table 1.1):

1. A complete state-of-the-art (SOTA) **study** of weakly supervised and unsupervised segmentation techniques. This task aimed to provide a clear understanding of the current SOTA in the field and to identify techniques to densify sparse point-level labels. Superpixel-based approaches and segmentation foundation models like SAM were considered.

2. Testing the studied techniques on several real underwater biology **benchmarks** to identify weaknesses and potential areas for improvement. During this task, we adapted and tested existing methods to know how they performed in challenging underwater scenes.

3. Design and implementation of a semantic **segmentation approach** with minimal supervision in labeling by taking advantage of foundation models. The system first augments the sparse point-level labels into dense mark labels, and second, it generates new masks by exploiting visual knowledge of foundation models.

4. **Evaluation** of the system in two ways: (1) dense augmentation accuracy and (2) the impact of automatically augmented labels in the training of a semantic segmentation model.

5. **Documentation** and intermediate presentations. Write a comprehensive report documenting the work carried out, including the methodology, results and a detailed review of the relevant literature. The report will also discuss the implications, limitations and possible future directions of the research. In addition, give several intermediate presentations to supervisors and colleagues during the project, encouraging feedback and discussion to improve the quality and rigour of the research.

Task	Jul	Sep	Oct	Nov	Dec	Jan
Study						
Benchmarks						
Segmentation Approach						
Evaluation						
Documentation						

Table 1.1: Gantt diagram representing the distribution of the tasks done along the months.

1.3 Context and tools

This work was developed in the Robotics, Computer Vision, and Artificial Intelligence group at the University of Zaragoza, within the Institute of Engineering and Research of Aragon (i3A). It serves as the starting point for my PhD thesis and will be further developed in future research. Also it is a continuation of a previous work done during a 4-month internship in this group that concluded with the publication of an article in a national journal, *Revista Iberoamericana de Automática e Informática Industrial* (RIAI), which is about automatic scene understanding in underwater environments utilizing depth estimation and unsupervised segmentation techniques [8].

Additionally, this project is part of a multidisciplinary collaboration with The Byrnes Lab, a marine ecology laboratory at the University of Massachusetts, Boston (UMass Boston). The lab provided point-labeled underwater image data, primarily focusing on coral and flora. Our system could support their research by enabling efficient analysis and interpretation of complex underwater images, eliminating the need for exhaustive and costly labeling efforts.

The main programming language used for this project was `Python`, using `Visual Studio Code` as development environment and Ubuntu 22.04 LTS as operating system. `Conda` (version 24.7.1) was used as package and Python environment management system. The experiments

were run using widely known machine learning libraries such as `Numpy` (base version 1.26.4), `OpenCV` (version 4.7.0) and `PyTorch` (version 2.5.0).

The hardware used to run the experiments was a local machine with a NVIDIA GeForce RTX 4090 GPU. `PyTorch` library leverages GPU acceleration to significantly speed up computations, reducing execution times for the large models and datasets used during this master's thesis.

The project documentation was created using the widely-used and well-established typesetting system LaTeX, with editing done on `Overleaf`.

In Appendix A.1, we reference the GitHub repository hosting the software tool developed as part of this work.

1.4 Project structure

The master's thesis documentation is divided into the following chapters:

- Chapter 1 of the project, the introduction, provides an overview of the context, tools, objectives, and tasks of the project.
- Chapter 2 provides an overview of the key background concepts that form the basis of this project. The first section explores foundation models, with a particular focus on those designed for visual recognition, discussing their capabilities and relevance to the field. The second section delves into semantic segmentation, tracing its evolution from classical approaches to SOTA techniques, with an emphasis on the role of foundation models in this domain. The chapter concludes with a discussion on weakly supervised learning, highlighting the challenges it addresses and introducing label propagation methods as a key strategy in this context.
- Chapter 3 focuses on the proposed approach for label augmentation. It starts by defining the problem and introducing the explored methods used as starting point and for comparison. The chapter then provides a detailed description of the developed approach, highlighting its design and explaining the functionality of its individual modules. Finally, the chapter presents an example application where the augmented labels are used to train a deep learning semantic segmentation model for a specific wildlife monitoring task.
- Chapter 4 details the experiments conducted throughout the project. It begins by presenting the experimental setup, including the datasets used and the evaluation metrics applied. Next, the chapter presents the results of our approach, highlighting the performance and limitations of label augmentation, as well as the impact of automatically augmented labels on the training of a semantic segmentation model. Finally, it discusses how the placement of initial sparse points influences the quality of the augmented labels.
- Chapter 5 presents the conclusions drawn from the obtained results, as well as a discussion on the limitations of the work and possible future steps to be taken.

Chapter 2

Related work and Background

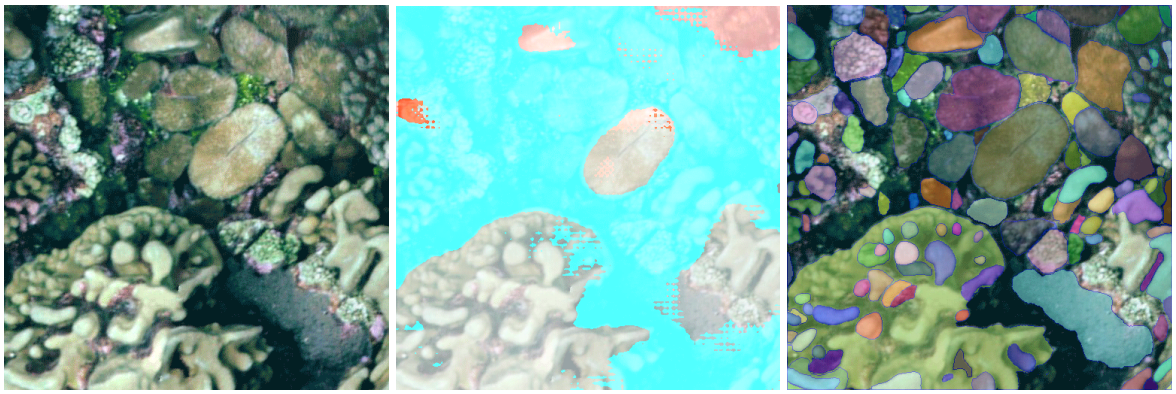
2.1 Foundation models

The rapid advancements in artificial intelligence can largely be attributed to the emergence of foundation models. These models, pre-trained on massive datasets through techniques such as self-supervised learning, are designed to capture general patterns and representations. As a result, they are able to tackle a wide range of tasks with minimal fine-tuning or task-specific adaptations. While foundation models were first popularized in natural language processing with large language models [9, 10, 11] their success has inspired similar approaches in computer vision.

Following the success of transformers in natural language processing, we have seen the emergence of powerful generic visual recognition models built on Vision Transformers (ViTs) [12]. Unlike traditional Convolutional Neural Networks (CNNs) [13], which are inherently limited to local information, ViTs use attention mechanisms across small patches of pixels, allowing them to capture long-range dependencies and global context within the image, crucial for understanding complex scenes [12]. This makes them especially effective for learning from large-scale visual data. Notable examples include CLIP [6], which aligns images and text in a shared embedding space, allows general-purpose visual understanding without the need for task-specific fine-tuning. Another example is DINOv2 [5], a self-supervised model that trains on images without using labels. It achieves strong performance in tasks like object detection, segmentation and depth estimation, demonstrating the capability of Vision Transformers to learn high-quality image representations. Finally, SAM and SAM2 built on ViTs for universal segmentation tasks. SAM supports segmentation of objects in images using flexible user prompts like points, boxes, or masks [7], while SAM2 extends these capabilities to video data, offering improved efficiency and segmentation accuracy for both images and videos [14]. In this work, we leverage foundation models to improve weakly supervised segmentation models focusing on wildlife monitoring scenarios.

2.2 Dense segmentation

We refer to dense segmentation as the task of partitioning an image into regions of similar characteristics, such as color, texture, or intensity. Over the years, this field has evolved significantly with various methods contributing to its advancement. Traditionally, unsupervised methods have played a significant role in image segmentation. Superpixel techniques [15, 16, 17, 18] group neighboring pixels with similar color and texture properties. However, these



(a) Query Image. (b) DINOv2. (c) SAM2.

Figure 2.1: Semantic segmentation comparison between (b) DINOv2 and (c) SAM2 on a representative (a) underwater image. Both models operate without any additional input beyond the image itself. DINOv2 extracted features can be used for a variety of tasks, including semantic segmentation. It groups similar patches into the same semantic category, even between distant areas of the image. In contrast, SAM2's generates much sharper and precise segmentations, but it does not semantically relate segments, treating each segment independently.

methods often struggle to accurately capture complex object boundaries and can be sensitive to noise and variations in image appearance.

CNNs have played a key role in semantic segmentation. Early approaches involved extracting features from the image using CNNs and then feeding these features to a fully connected layer for pixel-wise classification. However, this approach had limitations, primarily due to the loss of spatial information. Fully Convolutional Networks (FCNs) [4] addressed this by replacing fully connected layers with convolutional layers. This allowed FCNs to process images of arbitrary size and generate dense pixel-wise predictions. Examples of FCN-based models include U-Net [19], a popular architecture for medical image segmentation known for its encoder-decoder structure with skip connections that preserve fine-grained details, and the DeepLab [20, 21, 22] family of models, which incorporate spatial pyramid pooling (SPP) to capture multi-scale information effectively.

Recently, transformer-based architectures have revolutionized semantic segmentation. Some examples of such architectures include SegFormer [23], which employs a hierarchical transformer encoder and a lightweight MLP (Multi-Layer Perceptron) decoder for efficient segmentation, and Swin Transformer [24] which introduces a hierarchical and window-based approach to efficiently process high-resolution images. As mentioned in previous section, ViT's ability to capture global relationships within an image has inspired the development of numerous vision foundation models, some of which have shown remarkable performance in semantic segmentation tasks. Among these, DINOv2 [5], SAM [7], and subsequently, SAM2 [14] have emerged as strong candidates. Both models extract image features, with DINOv2 providing general-purpose features for various tasks, while SA,2 specializes on segmentation, generating segment candidates over the image.

While models like SAM2 and unsupervised techniques such as superpixels stand out at generating dense segmentation of images based on visual similarity, they do so without establishing semantic relationships between the segmented regions (Figure 2.1). In this work, we focus on **semantic segmentation** task which additionally classifies each generated segment into a predefined set of categories. To address this task, we build an approach that leverages these mentioned segmentation models (SAM, superpixels, DINOv2) to generate pseudo-labels and, subsequently, train a semantic segmentation model.

(a) Sparse point-level (b) Bounding box (c) Text prompts

Figure 2.2: Examples of scarce initial labels that could serve as starting points for propagation using weak supervision models to get our final generated dense masks.

2.3 Weakly supervised learning

Weakly supervised learning focuses on training models with limited or imprecise annotations rather than exhaustive ground truth labels. This approach aims to balance model performance with reduced labeling effort, often through strategies like **label augmentation**. Label augmentation is a technique for transforming sparse annotations into dense predictions. For example, in video action recognition, sparse annotations on select frames can be extended across entire action clips [25, 26]. Also in videos a single annotated mask in one frame can be propagated across subsequent frames to generate dense segmentations for the entire sequence [14]. Among its applications, weakly supervised learning has been particularly impactful in semantic segmentation, where the goal is to produce accurate dense pixel-level predictions from limited or sparse annotations.

Weakly supervised semantic segmentation (WSSS) has been studied in the image domain, with notable approaches including methods that leverage point-level supervision [27, 28], methods that utilize image-level supervision to learn pixel-level semantic affinities [29], and methods that leverage generative adversarial networks (GANs) [30] that add large fake visual data to force real data to be closer in the feature space, improving multi-class pixel classification [31]. Also, it has been studied in 3D point cloud semantic segmentation, where annotation challenges are compounded by the large data size and intricate geometric structures. These approaches achieve competitive performance compared to fully supervised methods while significantly reducing annotation costs [32, 33].

In the context of wildlife scenarios, acquiring pixel-level annotations for wildlife imagery is even more challenging due to variations in lighting, occlusions, and the dynamic behavior of wildlife. Consequently, it is common to encounter datasets scarce labeling, where annotations are limited to sparse point-level labels, bounding boxes, text prompts, among others (Figure 2.2. Current methods for WSSS in underwater environments often rely on label augmentation techniques from point-level labels. These methods aim to obtain a dense segmentation of the images based on a limited number of sparse point-level annotations.

CoralSeg [34] utilizes a multi-level superpixel segmentation strategy for label propagation. This approach iteratively applies superpixel segmentation with decreasing numbers of superpixels, starting with a high number of small superpixels to capture fine details and gradually merging them in subsequent iterations. This multi-level strategy effectively addresses the limitations of single-level approaches, which can suffer from either excessive unlabeled regions or insufficiently accurate shape representation. Raine et al. [35] introduced a novel approach that optimizes superpixel centers using a custom loss function. This loss function considers both pixel feature similarity and the presence of conflicting class labels within each superpixel. By

minimizing this loss, the method generates superpixels that closely conform to coral boundaries and minimize the inclusion of conflicting class labels. This approach utilizes the encoder of the Superpixel Sampling Networks (SSN) [36] to extract informative features for superpixel generation. Furthermore, their recent work [37] explores a human-in-the-loop approach to improve annotation efficiency, demonstrating significant gains in segmentation accuracy in extremely sparsely labeled images.

In our preliminary work [8], SAM2 demonstrated superior accuracy in generating high-quality segmentation masks compared to superpixel-based methods. SAM2's ability to produce precise object boundaries and coherent masks significantly outperformed superpixel techniques, even when applied in challenging scenarios. This precision was evident when expanding individual point-labels into dense segmentations, addressing the challenge of aligning segmented regions with actual object shapes. Superpixel-based methods, on the other hand, offer the advantage of segmenting unlabeled areas by leveraging feature-based similarities. While this is conceptually appealing, these techniques struggle when dealing with extremely sparse point annotations. The reliance on over-segmentation (where object regions are segmented too broadly, losing the detailed shape of the object) often leads to fragmented and confusing results, as the generated superpixels fail to capture the complexity of object shapes in such scenarios.

In this work, we augment point-level annotations into dense segmentation masks by propagating labels based on spatial and feature-based similarities.

Chapter 3

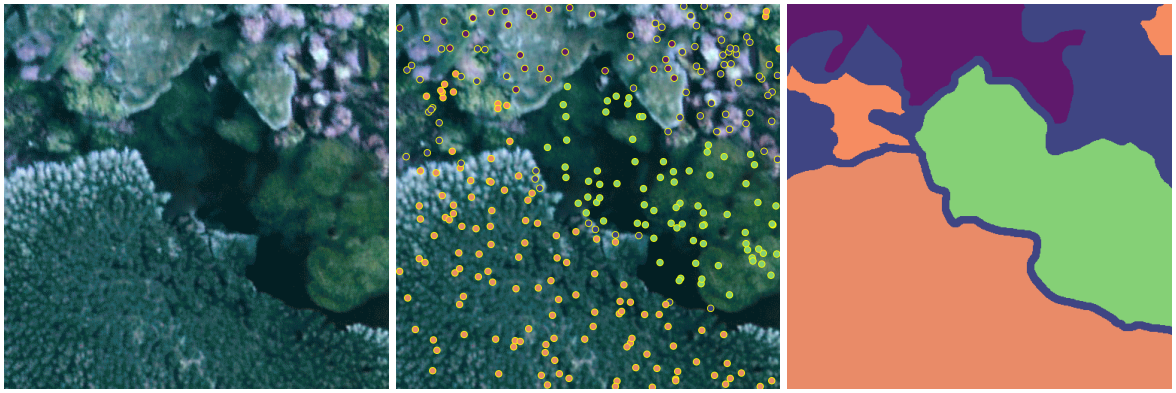
Label augmentation

This chapter describes a novel point-level label augmentation approach leveraging vision foundation models. By integrating the strengths of models like SAM2, this approach aims to improve the accuracy and coverage of dense segmentations generated from sparse point-level labels. The chapter begins with a formal problem definition and a discussion of baseline methods before detailing the proposed approach. The methodology is divided into distinct modules, including point label propagation and refinement. We conclude proposing an application for the augmented labels by using them as training data for a deep learning semantic segmentation model.

3.1 Problem definition

The goal is, given a set of sparse point-labels, to produce high-quality dense segmentation masks in an image and assign a semantic label to each mask (Figure 3.1). Formally, given an image I, and a set of point-label pairs $P_L = \{(p_1, l_1), (p_2, l_2), \ldots, (p_n, l_n)\}$ which each pair is formed by a pixel coordinate $p_i \in I$ and a ground truth (GT) semantic label $l_i \in L$.

The target is to generate a set of mask-label pairs $M_L = \{(m_1, l_1), (m_2, l_2), \ldots, (m_k, l_k)\}$, where each mask m_j is a set of pixels such that the union of all masks covers the entire set of image pixels, i.e., $\bigcup_{j=1}^{k} m_j = I$, and $l_j \in L$.



(a) Image. (b) Sparse ground truth point-labels. (c) Dense semantic segmentation.

Figure 3.1: Example of the goal of our work: starting from (a) an image and (b) a set of sparse ground truth point-labels, we aim to generate a (c) dense semantic segmentation of the image that covers it entirely.

3.2 Approach

To address the challenge outlined in the problem definition, this section details the proposed methodology for label augmentation.

3.2.1 Context: related works

The proposed approach is built upon two dense segmentation methods for transitioning from sparse point-level labels to dense semantic segmentations. These methods serve as starting point for the development of our framework.

Superpixel-based propagation (SPX) [35]

This method uses the point label aware superpixel approach introduced by Raine et.al [35], which generates superpixels optimized for pixel feature similarity. It formulates the superpixel generation process as a clustering problem, where the location of each superpixel center is optimized through a loss function. The loss function combines two terms:

1. **Distortion Loss:** This term encourages the formation of superpixels by grouping pixels based on feature similarity. Pixel features are extracted using a feature extractor encoder, such us the SSN [36] or ResNet-18 [38] encoders.

2. **Conflict Loss:** This term ensures that each superpixel contains pixels from a single class by penalizing superpixels that include conflicting point-labels. The conflict loss minimizes the overlap in fuzzy memberships of labeled pixels with different classes, resulting in superpixels that better fits to object boundaries and are more suitable for label propagation.

This method works with pre-trained feature extractors (e.g., ResNet-18 from ImageNet [39]) and performs comparably well to approaches that fine-tune or train on specific datasets.

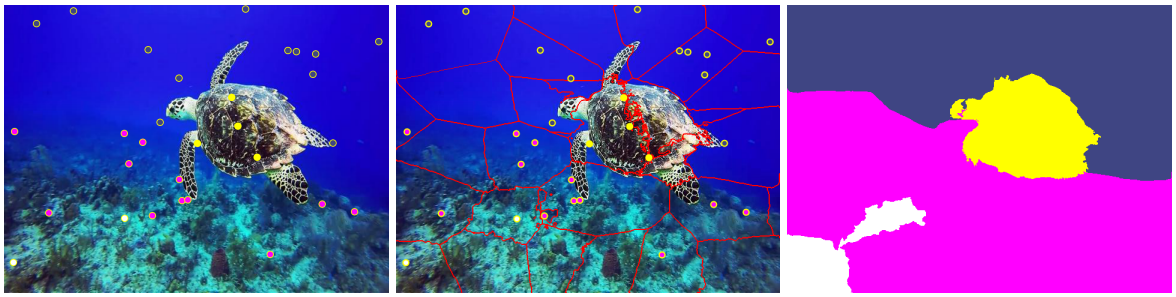
Once it creates the superpixel map, ensuring non conflicting labels, the label propagation is performed in two stages. First, for each superpixel containing at least one point label, it propagates that label to the superpixel. Second, for each unlabeled superpixels -i.e. those superpixels that do not contain any labeled point-, it compares their features with respect to the features of labeled superpixels, and it assigns the class of the most similar labeled superpixel (Figure 3.2). This ensures that every pixel in the image receives a label, resulting in a fully dense semantic segmentation, consistent with the original sparse annotations.

While this approach effectively propagates labels in certain scenarios, it can struggle with extremely sparse annotations, often leading to over-segmentation and a higher rate of false positives.

SAM2 segmentation [14]

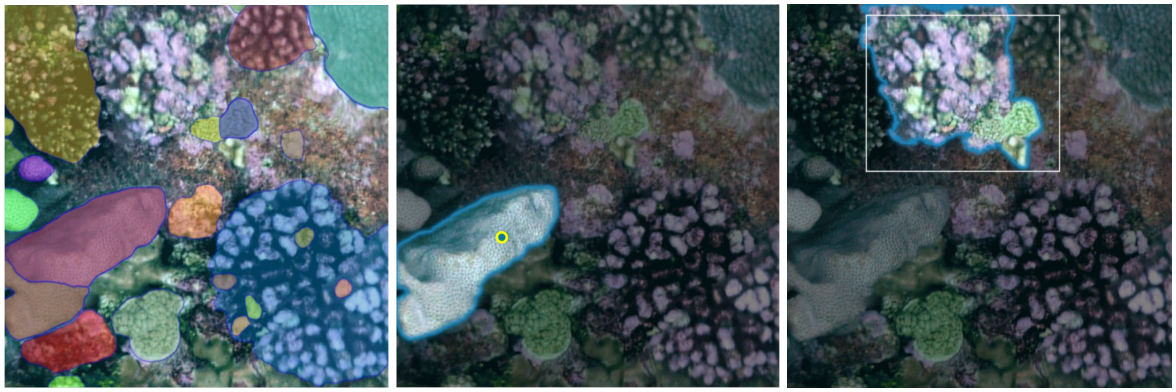
SAM2 is the state-of-the-art for image and video segmentation. It can generate segmentation masks in two ways: (1) automatically using a `SAM2AutomaticMaskGenerator` object (Figure 3.3.a), or (2) guided by specific input queries, by loading the image into a `SAM2ImagePredictor` object, which calculate the image embedding and allows mask prediction for different queries, such as points (Figure 3.3.b) or bounding boxes.

While SAM2 provides high accuracy in defining object boundaries, it is limited by its reliance on query-specific input, meaning it can only segment regions associated with the labeled



(a) Query image and point-labels. (b) Superpixels generation. (c) Label propagation.

Figure 3.2: Point label aware superpixel approach: starting from (a) the query image and a set of point-labels (where each color denotes a different semantic category), it creates (b) a superpixel map generation ensuring non-conflicting labels, and it finally performs (c) label propagation across the superpixels.



(a) Automatic segmentation. (b) Segmentation from query point. (c) Segmentation from query bounding box.

Figure 3.3: SAM2 segmentation usage modes: (a) automatic segmentation of the entire image, where SAM2 generates all possible masks without external queries, and query-based segmentation, where SAM2 predicts masks guided by specific inputs, such as (b) points or (c) bounding boxes.

points. SAM2 can generate masks independently, but determining the correct label for such segments without guiding queries remains an unresolved challenge.

3.2.2 Overview

Our aim is to produce high-quality dense segmentation masks in an image and assign a semantic label to each mask, starting from a set of sparse point-labels. To achieve this goal, we build an approach upon the two previous explained works (SPX and SAM2). Our pipeline is divided into two main stages, represented by separate modules: Label Propagation and Refinement (Figure 3.4). The system takes as input an image and a .csv file containing the position and semantic label of the sparse points, and outputs a dense segmentation of the image through two different alternatives. Here is a short description of each module:

1. **Label Propagation:** This module processes the raw image and sparse point-level annotations. Two parallel strategies are used:

    - SPX propagation: Produces a fully labeled semantic segmentation without unlabeled pixels. While this segmentation is complete, it lacks the precision of SAM2 expansions and serves as an intermediary output for subsequent processing in the next module.

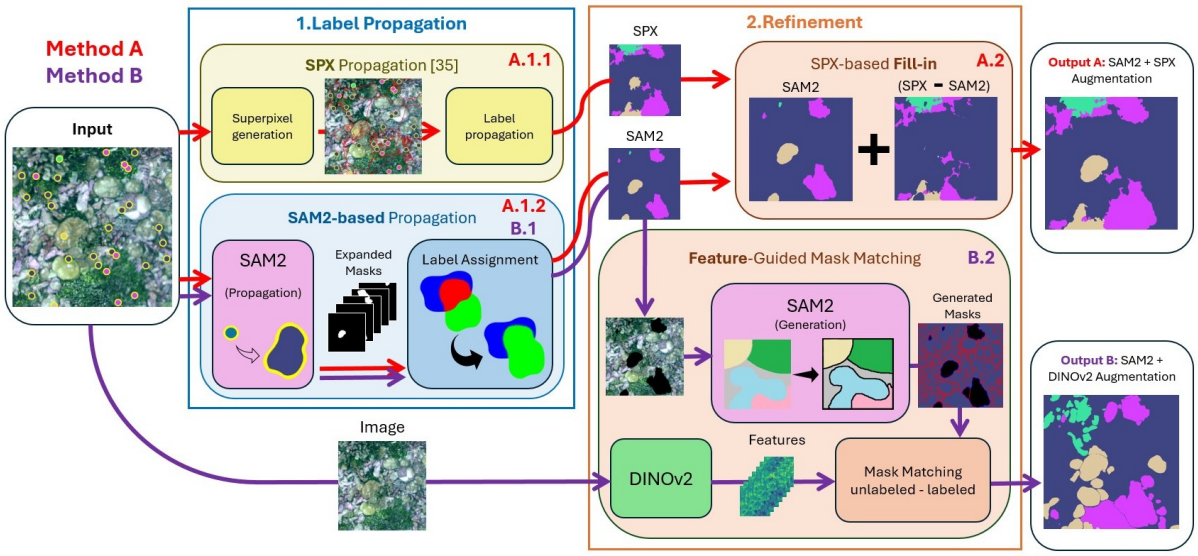


Figure 3.4: Overview of the proposed label augmentation framework. The system takes an image and sparse point-labels as input, exploring two distinct methods for label augmentation through two main stages: Label Propagation and Refinement. In label propagation stage, labels are propagated using: **A.1.1** SPX propagation and **A.1.2**/**B.1** SAM2-based propagation. In refinement stage, the system addresses unlabeled regions in two ways: **A.2** SPX-based fill-in, which uses **A.1.1** to fill gaps in **A.1.2**, and **B.2** feature-guided mask matching. The latter method generates new SAM2 masks based on **B.1**, and leverages DINOv2 features to compare these new masks (unlabeled) with the initial ones (labeled), assigning labels based on feature similarity. So the **two explored methods for label augmentation are**: **MethodA = (A.1.1 + A.1.2) + A.2 and MethodB = B.1 + B.2**.

- SAM2-based propagation: Propagates each labeled point into a mask, which are then merged to resolve overlapping regions and assigns the label of the point to the generated mask. It produces accurate but incomplete segmentations with some unlabeled areas which are passed to the next module for refinement.

2. **Refinement**: In this module, we try to assign a label the unlabeled regions of the SAM2-based propagation output using two distinct strategies:

   - SPX-based fill-in: The SPX-based propagation from the previous module is used to fill the unlabeled regions in the SAM2 masks. This results in the first alternative of our system.
   - Feature-guided mask matching: This method enhances the SAM2 propagation by generating candidate masks over a modified image (by blacking out the expanded masks). Features are extracted from the original image using DINOv2, and each unlabeled mask is assigned to the label of the most similar labeled mask in the feature space. This results in the second alternative of our system.

In Section 3.2.3, we detail Label Propagation module, while Section 3.2.4 explains Refinement module.

3.2.3 Module 1: Label Propagation

The Label Propagation module is responsible for propagating the sparse point-level labels into a more complete semantic segmentation by leveraging two methods: SPX propagation and SAM2-based propagation. Each method has distinct features that contribute to generating

different dense label segmentations. The module takes as input the image and the point-level labels and performs the following steps.

SPX propagation (A.1.1)

This module section which is part of our MethodA (Figure 3.4) utilizes the superpixel propagation approach explained in Section 3.2.1. This generation will be used later in Label Refinement module of the Method2 to fill the gaps of the SAM2-based propagation.

SAM2-based propagation (A.1.2 and B.1)

The SAM2-based propagation submodule which is part of our two methods (Figure 3.4) utilizes SAM2 to generate masks from query points and assign the point label to the mask. This decision is based on findings from preliminary studies made during my internship, which showed that propagation is faster than generating all masks automatically and then identifying those containing the points. Also, automatic mask generation could skip producing detailed masks for objects with assigned points. Thus, this approach involves using SAM2 to expand each point label into a segmentation mask and then resolving overlaps among these masks as part of the process.

Expanding point-labels. First, we generate a mask with SAM2 from each of the query points. In this way, we run as many inferences of SAM2 as point-labels we have. For each inference, we follow the next steps:

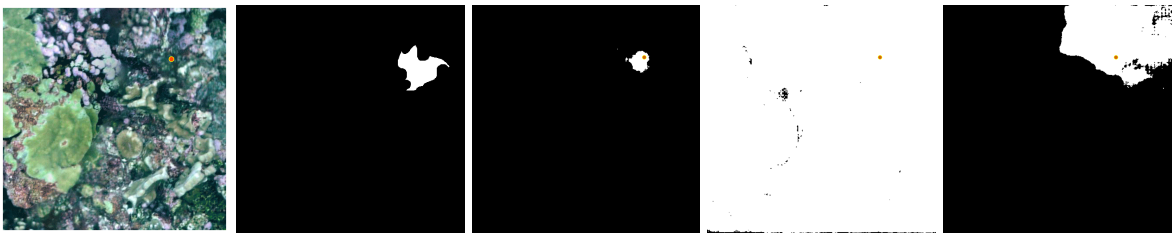
1. **Mask prediction:** By default, all pixels are considered as "background" before the mask prediction. For each labeled point, three candidate masks are predicted for the given point label using SAM2 with `multimask_output` enabled. The multimask approach addresses the inherent ambiguity of a single-point prompt by producing multiple segmentation options. Sparse points labeled as background are not expanded, both to improve computational efficiency and because, they are already treated as background.

2. **Mask selection:** Among the three masks, the most suitable one is selected using a weighted scoring scheme based on the following:

   - **Confidence (s):** Provided by SAM2 for each mask, it represents the model's confidence in the prediction. However, in practice, we observed that a high confidence does not always lead to accurate expansions. Therefore, additional metrics are considered to ensure better mask selection.
   - **Compactness (c):** A measure of the geometric regularity of the mask, calculated as:

$$c = \min\left(\log\left(1 + \frac{2\sqrt{\pi A}}{P}\right), 1\right), \tag{3.1}$$

     where A is the mask area and P is the perimeter. Higher values indicate more compact masks, approaching a shape with minimal perimeter for a given area. This formulation penalizes irregular and and highly scattered shapes but avoids over-rewarding highly compact shapes, such as perfect circles.



(a) Query. (b) GT mask. (c) Candidate 1. (d) Candidate 2. (e) Candidate 3.

Candidate	Confidence (s)	Compactness (c)	Size penalty (p)	**Score**
Candidate 1	0.0361	0.4114	-0.0081	**0.4394**
Candidate 2	0.1611	0.4065	-1.5686	-1.0010
Candidate 3	0.2559	0.2542	-0.1816	0.3285

Figure 3.5: Mask selection example. (a) Query image with a point label to be propagated, (b) ground truth (GT) mask, and (c)-(e) mask candidates predicted by SAM2. The table shows the scores used to select the best candidate mask. Although (c) has the lowest confidence score, its high compactness and low size penalty make it the selected mask, as it is also closest to the ground truth. (d) demonstrates a poor prediction, nearly covering the entire image and being heavily penalized by the size penalty term p.

- **Size Penalty (p):** A penalty applied to discourage masks that are too small or too large relative to the image size. It is defined as:

$$p = n + p_s + p_l, \tag{3.2}$$

where n is the normalized mask area, defined as the ratio of mask pixels to the total number of pixels in the image, and

$$p_s = \begin{cases} n^4, & \text{if } n < 0.001 \\ 0, & \text{otherwise} \end{cases}, \qquad p_l = \begin{cases} (n-0.4)^4 & \text{if } n > 0.5 \\ 0 & \text{otherwise} \end{cases}, \tag{3.3}$$

are the penalties for too small masks (less than 1% of the image) or too large masks (more than 50% of the image), that ensure the mask size is reasonable. These coverage percentages were selected manually after examining numerous examples of expanded point-labels in coral reef images, which indicated that these are reasonable minimum and maximum sizes, specially considering that the mask is generated from a single point.

The final weighted score for each mask is calculated as:

$$\text{Score} = w_s s + w_c \log(1+c) - w_a p, \tag{3.4}$$

where w_s, w_c and w_a are weights for s, c, and p, respectively. The values of these weights were fine-tuned through qualitative testing on images from the UCSD Mosaics [40] dataset, focusing on scenarios where the three masks predicted by SAM2 varied significantly in size and shape. In certain cases, some masks covered almost the entire image, throwing off label propagation. To address this, we adjusted the weights to prioritize smaller masks, which despite having lower confidence, to better preserve the mask selection process. The mask with the highest weighted score is selected as the final expansion for the point label (Figure 3.5).

Label assignment. Once all masks are generated from the point-labels, we assign a label to each generated mask. Some masks may overlap with others of different labels. To resolve these conflicts, the algorithm first identifies the overlapping regions by checking for intersections between the bounding boxes of all mask pairs. The intersecting pixels are then grouped, and the masks involved are recorded.

Next, the algorithm assigns a label to these overlapping regions based on the distribution of point-labels of the involved masks. If one semantic label has the majority of point-labels in the overlap, that label is assigned to the entire region. If no majority exists, the semantic label of the closest point-label to the centroid of the overlap region is used for all pixels in that region.

Finally, pixels that do not overlap are added directly to the output mask with their original labels. This ensures that all pixels are labeled, and conflicts in overlapping areas are resolved by considering both the majority of point-labels and the proximity to the closest point-label. The label assignment algorithm is available in Appendix A.2. Example results for three different overlap solving cases are shown in Figure 3.6.

Also, Figure3.7 illustrates some SPX and SAM2-based propagation examples for comparison.

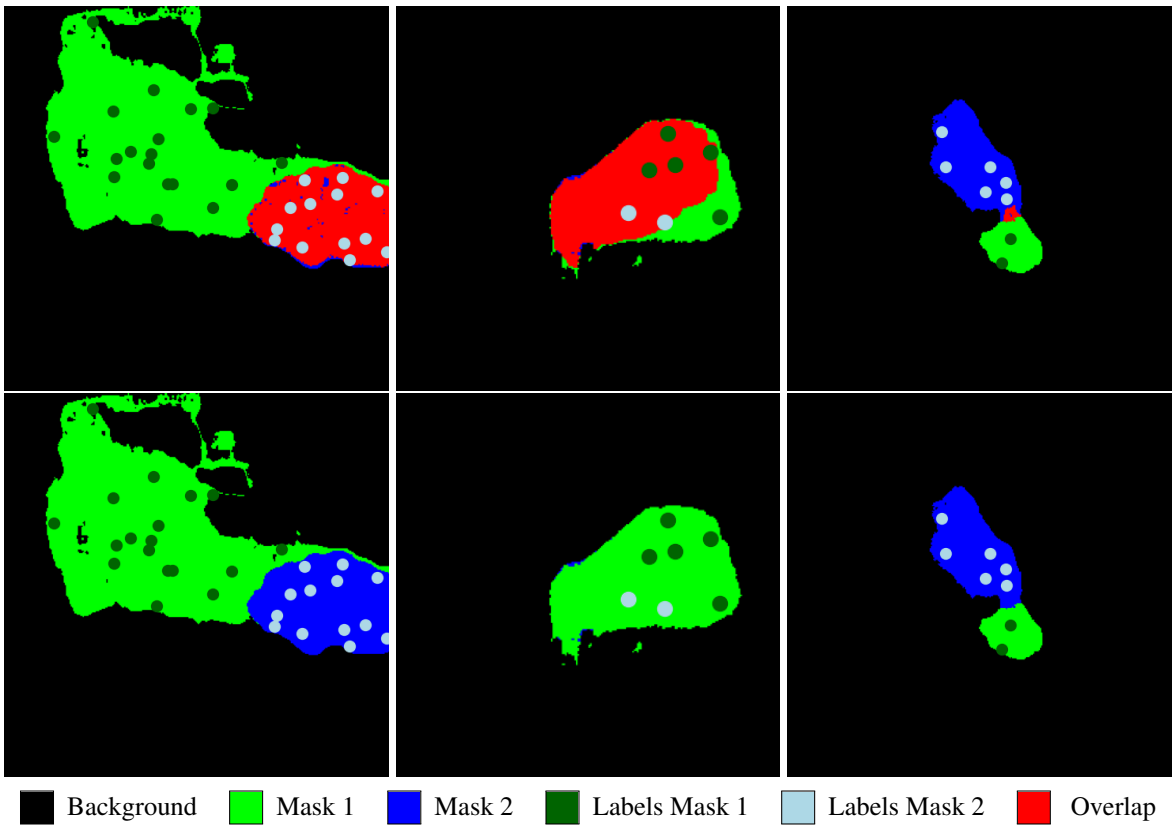


Figure 3.6: Overlap Solving. Top row: examples of overlap between two masks. Bottom row: resolved overlaps. In the first two cases, resolution is based on the majority label within the overlap area, determined by the point-labels inside it. The third case is resolved by proximity to the point-labels.

3.2.4 Module 2: Refinement

The Refinement Module addresses the unlabeled pixels remaining after the SAM2 propagation. While SAM2 is effective at generating high-quality segmentations, it has a limitation: regions without masks will remain unlabeled. The objective of this module is to label these unlabeled

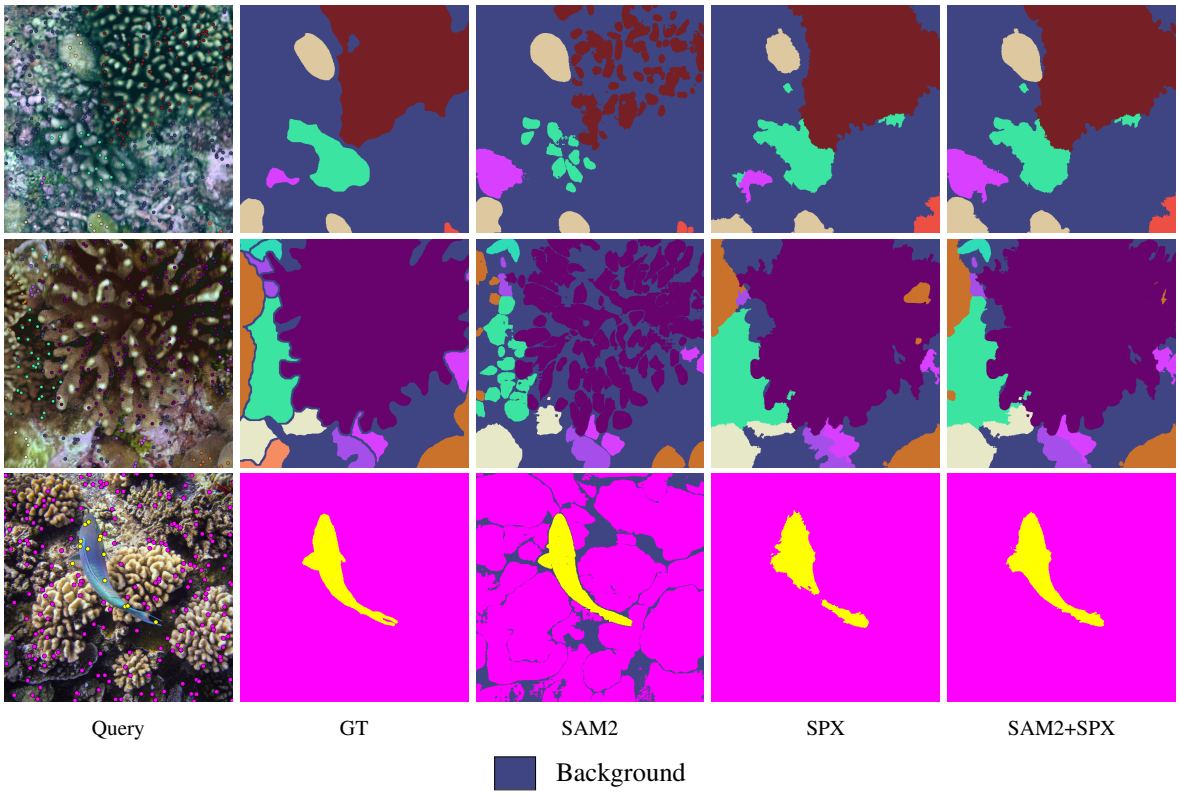


Figure 3.7: Successful examples where SPX+SAM improves coverage, addressing gaps in the SAM2 propagation. Each row shows, from left to right: the query image, ground truth (GT), SAM2 propagation, SPX propagation, and SAM2+SPX augmentation.

pixels as accurately as possible. To achieve this, two distinct approaches are explored to handle the unlabeled areas, aiming to improve the completeness and precision of the final semantic segmentation while maintaining the consistency with the existing labels.

SPX-based Fill-in (A.2)

This submodule is part of the MethodA and it is charge of refining SAM2 propagation by filling in unlabeled regions using the SPX propagation from Section 3.2.3 (Figure3.4). Unlabeled pixels in the SAM2 output are directly replaced with the corresponding labels from the SPX propagation. In mathematical terms,

$$\text{SAM2}_{\text{refined}} = \text{SAM2} \cup (\text{SPX} - \text{SAM2}). \tag{3.5}$$

This straightforward filling process complements SAM2's accurate boundary delineations by utilizing the SPX propagation's dense coverage, aiming to provide a more complete labeling of the image. This approach enhances coverage in scenarios where SAM2 expansions are small and leave significant unlabeled regions (Figure 3.7), however, it also introduces a drawback. In some cases, specially with a low number of sparse point-labels, the SPX propagation over-segments -i.e.the segmentation covers more area than it should- certain shapes that SAM2 captures accurately, reducing the accuracy of the combined segmentation. This is evaluated deeply in Section 4.2.1 (Figure 4.1).

To solve this, the next approach, focuses on feature space matching using DINOv2 and SAM2 to make better use of SAM2's accuracy.

Feature-Guided Mask Matching (B.2)

This alternative used in MethodB improves SAM2-based label propagation strategy by combining the automatic mask generation capability of SAM2 with feature information from DINOv2. In this way, this approach labels previously unlabeled regions by generating candidate masks and assigning labels based on semantic similarity in the feature space (Figure3.4).

Generating new masks with SAM2 The first step involves isolating the unlabeled regions from the SAM2 propagation described in Section 3.2.3. To ensure subsequent operations focus exclusively on the unlabeled areas, a modified version of the original image is created by applying a binary mask that blacks out all labeled regions by the SAM2 propagation (Figure 3.8.c).

The modified image is then processed by SAM2, which enables the generation of candidate masks without any query. Figures 3.8.d to 3.8.f show the difference between generating the masks, using the blacked-out image and using the original image. We can see that, using the blacked-out image we obtain new generated masks that align better with the original predicted masks. In Figures 3.8.e and 3.8.f, the yellow squares highlight areas where no mask was generated, even though these regions correspond to previously expanded masks. Since masks are generated directly on the original image, SAM2 does not produce masks in these areas. In the same way, the pink squares indicate discrepancies in the mask boundaries. This misalignment occurs because generating masks on the original image resulted in different boundaries compared to when the blacked-out image was used.

In conclusion, we chose to generate the new masks using the previously expanded masks, leveraging both SAM2's prediction and automatic mask generation capabilities. This approach ensures better alignment between the new candidate masks and the expanded masks. It also ensures that all expanded point-level labels are used in the process. If we would generate the masks directly on the original image, relying only on SAM2's automatic mask generation without using the prediction part, some point-level labels could be lost, as certain masks that should include these labels might not be generated.

DINOv2 feature extraction After generating new masks that cover the entire image, the next step is to assign labels to these masks. To do this, we use DINOv2 [5] to extract feature embeddings from the original image. DINOv2 generates an embedding with a fixed feature dimension of 1536 for each patch of 14×14 pixels (e.g., 37×37 for a 512×512 input image). These embeddings are then upsampled using bilinear interpolation to match the original image size (Figure 3.9). For each SAM2-generated mask, whether it is one of the original predicted masks expanded from labeled points or one of the newly generated masks, we calculate a mean embedding feature vector by averaging the embeddings of the pixels within the mask's region, resulting in a feature vector per mask of 1536 dimensions. This process ensures that both predicted and newly generated masks are consistently represented in the feature space, enabling comparisons between labeled and unlabeled masks.

Label Assignment via Embedding Similarity Once each mask has its corresponding embedding, we iterate over all unlabeled masks to assign them a label. For each unlabeled mask u, we calculate the cosine similarity between its embedding vector $\mathbf{e}_u$ and the embedding vectors $\mathbf{e}_l$ of all labeled masks l, defined as:

$$\text{Cosine Similarity}(\mathbf{e}_u, \mathbf{e}_l) = \frac{\mathbf{e}_u \cdot \mathbf{e}_l}{\|\mathbf{e}_u\| \|\mathbf{e}_l\|} \in [-1, 1]. \tag{3.6}$$

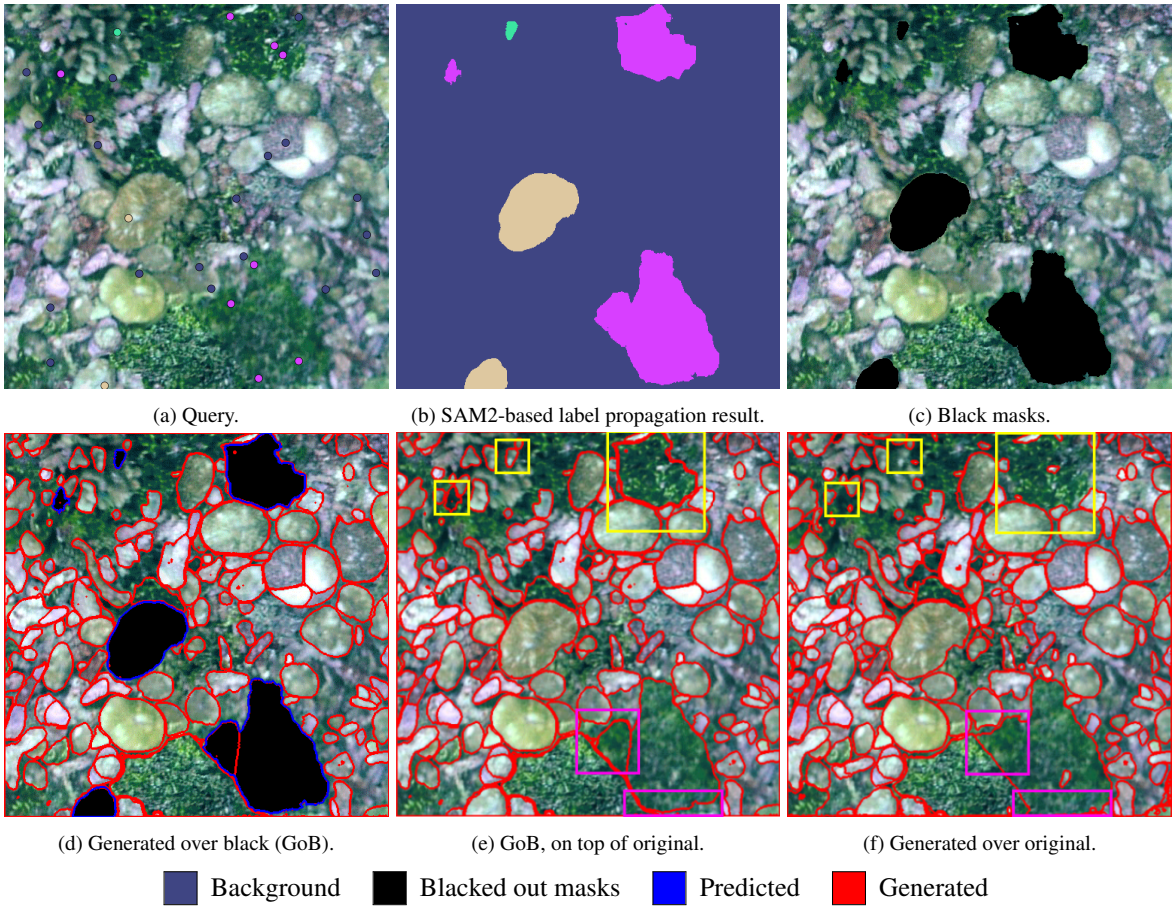


Figure 3.8: SAM2 mask generation comparison. (a) The original image with point-labels. (b) Predicted masks from SAM2 expansion. (c) Blacked-out version of the image, with labeled regions removed. (d) Generated masks over the blacked-out image. (e) Generated masks over the blacked-out image, but displayed on top of the original image. (f) Generated masks over the original image. Red boundaries represent newly generated masks, and blue boundaries in (d) represent the original predicted masks. In (e), the generated masks align well with the blacked-out areas in (d), ensuring better mask coherence. In (f), without generating new masks over the blacked-out image, the masks fail to align with the SAM2 expansion (pink squares) or are completely missing (yellow squares).

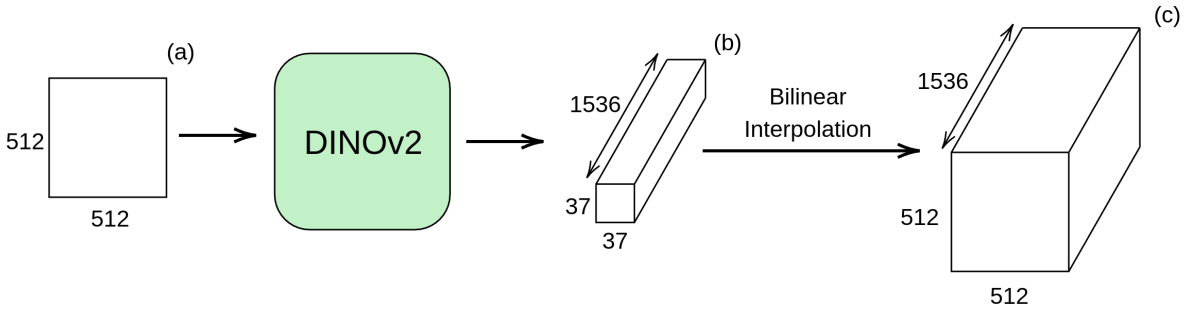
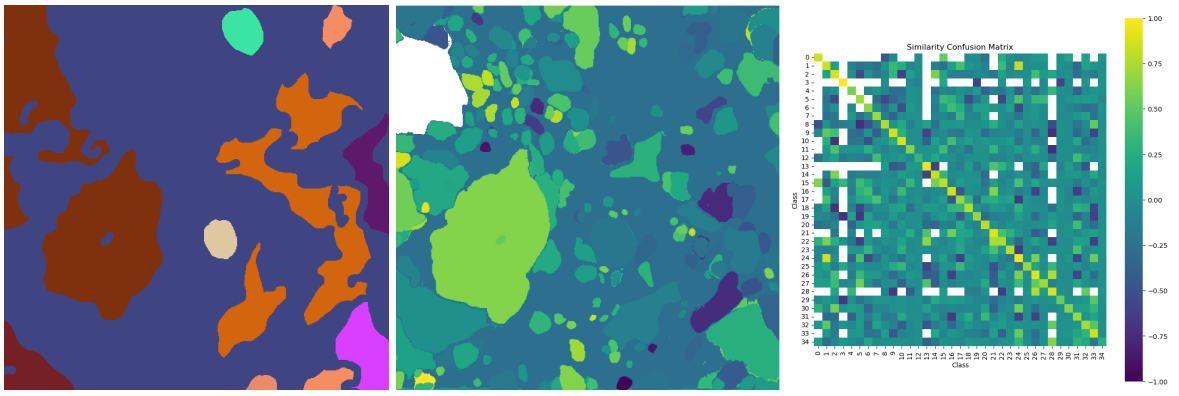


Figure 3.9: DINOv2 feature extraction. The original image is (a) passed through DINOv2 model, producing (b) a feature embedding with a fixed depth of 1536. The embeddings are then (c) upsampled using bilinear interpolation to match the original image dimensions.

To improve robustness, we introduce a threshold similarity $\tau = 0.6$, chosen manually in Appendix B.1. If the maximum similarity for a given unlabeled mask is below this threshold, the mask remains unlabeled and is treated as background. This step ensures that only masks with a strong correspondence to existing labeled masks are assigned a label, reducing the risk of incorrect assignments. Figure 3.10.c shows the cosine similarity confusion matrix among all the ground truth labels of the UCSD Mosaics [40] dataset. To get this plot, we compute for each image the mean feature vector of each mask and its cosine similarity with respect to all other mask features of the image (Figure 3.10.b). We may deduce from this plot that DINOv2 features can be useful to propagate labels among masks inside an image.



(a) GT semantic segmentation masks. (b) Sim. of reference (white) mask to others. (c) Mean similarity confusion matrix.

Figure 3.10: DINOv2 features capability. We show (a) the ground truth semantic labels of an image and (b) the cosine similarity of a reference (white) mask features with respect to all other mask features of the image. Lastly, we show (c) the average similarity confusion matrices of all the images. The visible diagonal indicates that the model correctly assigns high similarity to pairs of masks with the same label. However, some high similarity values off the diagonal reveal instances where masks of different labels are incorrectly considered similar. White elements represent pairs of semantic labels that never co-occur within the same image.

This label assignment method ensures that only masks with a strong similarity to labeled masks are assigned a label, reducing incorrect assignments due to weak correspondences. However, the effectiveness of this approach depends heavily on the quality of DINOv2 features. In these complex underwater environments with similar objects across different classes, the mean embeddings per class can lead to incorrect label assignments. Despite these challenges,

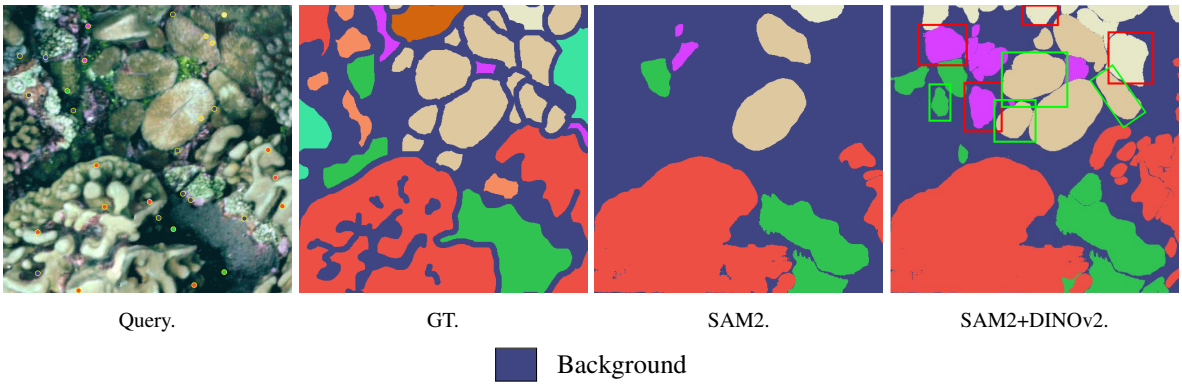


Figure 3.11: Example of SAM+DINOv2 label refinement. It shows new masks both correctly (green squares) and incorrectly (red squares) labeled.

SAM2+DINOv2 often succeeds in generating correctly labeled masks, improving semantic segmentation coverage in prior unlabeled areas. Figure 3.11 illustrates both the strengths and weaknesses of this method.

3.3 Application of the Augmented Labels

The scarcity of annotations in wildlife imagery datasets is a major challenge. Manual labeling requires significant time and effort, making it difficult to train machine learning models for tasks like semantic segmentation. Finding ways to address this limitation is essential. Our label augmentation approach helps to solve this problem by generating pseudo-labels that can be used to train models as we would do with ground truth labels. As result, our target is to prove that using only our generated pseudo labels we could train a model that achieves a similar performance than training it with all the dense ground truth labels. In this situation, we would achieve the same model performance but having reduced considerably the human effort for labeling.

For this purpose, we utilize SegFormer [23], a Transformer-based framework that combines the representational power of Transformers with lightweight MLP decoders. SegFormer's design makes it efficient and capable of learning multi-scale features, achieving strong performance across benchmarks. Its lightweight architecture allows for quick evaluation of the impact of pseudo-labels on model training. While our main contribution is the augmentation of labels, using them to train a model demonstrates their practical value and potential applications. The evaluation of the model training using augmented labels is presented in Section 4.2.3.

Beyond their role in training machine learning models, these pseudo-labels are valuable for researchers themselves, as they provide deeper insights into complex environments, supporting better analysis and measurement.

Chapter 4

Experiments

In this chapter, we present the experimental setup and results of the proposed label augmentation framework. We begin by outlining the experimental setup, including the datasets used and the evaluation metrics used. Next, we present the results of our approach, highlighting the performance and limitations of all methods explored. Later, we discuss how the placement of initial sparse points influences the quality of the augmented labels. Lastly, we analyze the performance of a semantic segmentation model trained uniquely from our generated pseudo-labels.

4.1 Experimental Setup

4.1.1 Datasets

We evaluated our framework using datasets that represent complex wildlife underwater environments.

UCSD Mosaics-. The UCSD Mosaics dataset [40] consists of 16 large mosaics of multi-species coral reefs, each with a resolution exceeding 10K $\times$ 10K. For our experiments, we used a cropped version of this dataset, where the mosaics were divided into 512 $\times$ 512 resolution images, resulting in 262,144 pixels per image. This yields a total of 3974 training images and 696 test images. The dataset includes 34 semantic classes, excluding the background class, and provides dense annotations for every pixel in each image.

SUIM-. The Semantic Segmentation of Underwater Imagery [41] (SUIM) dataset consists of 1525 training images and 110 test images, all with dense annotations across eight object categories: fish (vertebrates), reefs (invertebrates), aquatic plants, wrecks/ruins, human divers, robots, and sea-floor. These images were carefully captured during oceanic explorations and human-robot collaborative experiments, with annotations provided by human participants. For our experiments, we resized all images to a consistent resolution of 640 $\times$ 480.

Additionally, we also evaluated two sets of images provided by our collaborator laboratory, The Byrnes Lab at UMass: Boston. Both without dense annotations, so the evaluation of these images will be qualitative, and no direct comparison to ground truth is possible (Appendix B.2).

4.1.2 Evaluation Metrics

We use standard metrics for semantic segmentation: mean pixel accuracy (mPA) and mean intersection over union (mIoU). These metrics are used for evaluation in both label augmentation and in training a semantic segmentation deep learning model with the augmented labels.

Let TP_l, FP_l, and FN_l represent the true positive, false positive, and false negative pixels for semantic label l, respectively. Also, let L_{valid} represent the set of all semantic labels L, excluding the background label.

Mean Pixel Accuracy (mPA)-. This metric calculates the average proportion of correctly classified pixels across all L_{valid}. The pixel accuracy (PA) for each label l is defined as the ratio of correctly classified pixels to the total number of pixels belonging to that label in the ground truth,

$$\mathrm{PA}_l = \frac{\mathrm{TP}_l}{\mathrm{TP}_l + \mathrm{FN}_l}. \tag{4.1}$$

The mPA is the average of the PA for all semantic labels in L_{valid}:

$$\mathrm{mPA} = \frac{1}{|L_{\text{valid}}|} \sum_{l \in L_{\text{valid}}} \mathrm{PA}_l. \tag{4.2}$$

This metric averages pixel accuracy across all L_{valid}. Higher values indicate better segmentation performance. Since false positives are not considered in the calculation of PA_l, it is possible for a label l to achieve 100% mPA if all ground truth pixels for l are correctly segmented, even if some background pixels are incorrectly labeled as l. For example, if an image contains only one semantic label l in addition to the background class, and we predict all pixel images with label l, we will get $mPA = 100\%$.

Mean Intersection over Union (mIoU)-. This metric quantifies the overlap between the predicted segmentation and the ground truth for each semantic label in L_{valid}. The intersection over union (IoU) for label l is defined as the ratio of the intersection of the predicted and ground truth regions to the union of those regions,

$$\mathrm{IoU}_l = \frac{\mathrm{TP}_l}{\mathrm{TP}_l + \mathrm{FP}_l + \mathrm{FN}_l}. \tag{4.3}$$

The mIoU is the average of the IoU for all semantic labels in L_{valid},

$$\mathrm{mIoU} = \frac{1}{|L_{\text{valid}}|} \sum_{l \in L_{\text{valid}}} \mathrm{IoU}_l. \tag{4.4}$$

This metric measures both the model's ability to correctly identify pixels for each label and its ability to avoid misclassified pixels (false positives) and missed ground truth pixels (false negatives). A higher mIoU indicates better segmentation performance, providing a more comprehensive measure of model accuracy than mPA. Therefore, we will consider this metric as reference.

4.2 Results

In this section, we evaluate the performance of our label augmentation framework both quantitatively and qualitatively. We also test how effective are the augmented labels as training data for SegFormer [23].

In order to obtain quantitative metrics, we randomly sample the set of point-labels from the dense ground truth annotations of the UCSD Mosaics and SUIM datasets. We evaluated the framework using fixed number of sparse points, specifically with 30 and 300 points, to analyze how the number of sparse labels impacts the results.

Method nomenclature The label augmentation methods evaluated are:

- **SPX:** Point-aware superpixel propagation method [35] (Section 3.2.3). That is, this approach represents the output of A.1.1 submodule (Figure 3.4).
- **SAM2:** Our proposed point label propagation method based on SAM2 [14], which is explained in Section 3.2.3 and makes the A.1.2/B.1 submodule (Figure 3.4).
- **SAM2 + SPX:** Our proposed combination of SPX and SAM2 propagations with SPX-based Fill-in described in Section 3.2.4 that constitutes our MethodA.
- **SAM2 + DINOv2:** Our proposed refinement of SAM2-based label propagation using generated new masks and DINOv2 feature extraction, doing feature-guided mask matching. It is explained in Section 3.2.4 and represents our MethodB.

4.2.1 Label Augmentation

We first compare the performance of the four label propagation methods presented before on the UCSD Mosaics and SUIM datasets (Table 4.1).

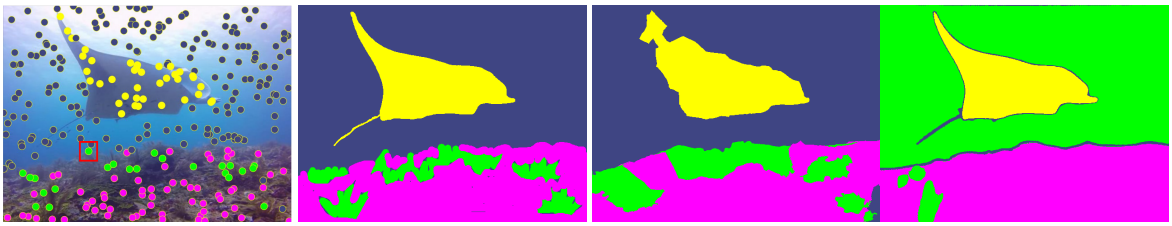
Table 4.1: Performance of label propagation approaches in UCSD Mosaics and SUIM. **Bold** and underlined means highest and second highest mIoU, respectively. Computational times are given as mean (±std) for all images.

Method	Ours	UCSD Mosaics			SUIM		
		mPA ↑	mIoU ↑	Time per Image (s) ↓	mPA ↑	mIoU ↑	Time per Image (s) ↓
300 Sparse Points							
SPX [35]	✗	90.90	71.96	2.02 (±0.10)	91.69	**82.51**	2.12 (±0.10)
SAM2	✓	85.62	**74.72**	2.26 (±0.89)	87.88	76.14	3.39 (±1.70)
SAM2 + SPX	✓	94.54	72.87	4.20 (±0.90)	94.77	78.02	5.52 (±1.70)
SAM2 + DINOv2	✓	88.23	66.45	17.79 (±3.47)	90.49	73.72	17.27 (±3.06)
30 Sparse Points							
SPX [35]	✗	67.86	45.55	1.76 (±0.10)	79.63	62.77	1.93 (±0.14)
SAM2	✓	52.11	49.30	0.83 (±0.35)	65.00	61.50	1.19 (±0.64)
SAM2 + SPX	✓	73.28	49.91	2.60 (±0.37)	85.85	**67.81**	3.16 (±0.60)
SAM2 + DINOv2	✓	61.31	**50.00**	13.58 (±0.90)	73.80	66.63	13.85 (±1.12)

When performing label propagation from 300 sparse point-labels, SPX performed well overall, achieving the highest mIoU on SUIM (82.51%) while keeping processing time low. This mIoU difference between SPX and the other methods is particularly remarkable in the

Table 4.2: mIoU per class in SUIM dataset for 300 sparse point-labels. Semantic classes in SUIM correspond to: Human divers (HD), aquatic plants and sea-grass (PF), wrecks and ruins (WR), robots (RO), reefs and invertebrates (RI), fish and vertebrates (FV), sea-floor and rocks (SR). While SPX and SAM2 share the best (**bold**) IoU results per class, SAM2+SPX finds the balance achieving the second-best (underlined) performance for most of all the classes. The images show an example of wrong propagation by SAM2 due to a PF point in the edge between the waterbody and the seabed (red square). This is an example of the importance of the location of sparse points that is discussed in Section 4.2.2.

	IoU per class							
Method	HD	PF	WR	RO	RI	FV	SR	mIoU
SPX [35]	73.98	**76.75**	**90.79**	79.83	**92.44**	74.56	**89.29**	**82.51**
SAM2	**83.68**	62.44	76.76	**86.06**	74.49	**77.25**	72.32	76.14
SAM2+SPX	79.16	66.66	80.28	82.28	83.45	76.81	77.55	78.02
SAM2+DINOv2	79.56	61.71	74.79	76.46	76.45	75.11	72.01	73.72



Query. GT. SPX. SAM2.

SUIM dataset which is carefully analyzed in Table 4.2 by way of an individual class performance comparison.

On the UCSD Mosaics dataset, SAM2 achieves the best mIoU (74.72%) and has a runtime similar to SPX. The SAM2+SPX method deliveres the highest mPA for both datasets, reaching 94.54% on UCSD Mosaics and 94.77% on SUIM, with solid mIoU results. SAM2+DINOv2 shows accurate mPA but had lower mIoU and required significantly more time due to the cost of generating new masks.

In the case of 30 sparse point-labels, SPX maintained a balance between performance and efficiency. SAM2, while being the fastest method and achieving competitive mIoU in UCSD Mosaics (49.30%), it provides low mPA in both datasets, mainly because of low coverage of the segmentation, leaving many pixels unlabeled. SAM2+SPX again achieves highest mPA across both datasets and performs nearly as well as SAM+DINOv2 in mIoU for UCSD, with best mIoU in SUIM dataset. In this case, SAM2+DINOv2 offered competitive accuracy, specially in mIoU, but the computational demand remains high.

Overall, SAM2 and SAM2+DINOv2 consistently achieve the lowest mPA values due to their tendency to leave large portions of the image unlabeled. This lack of coverage is penalized by mPA metric. Conversely, SPX-based methods (SPX and SAM+SPX) cover more areas, contributing positively to mPA. This could be because DINOv2 features sometimes group similar objects from different classes, leading to incorrect label assignments. Additionally, complex environments, with many similar objects across different classes, make it harder for the method to differentiate correctly. SAM2+DINOv2 is also, by far, slower than the other methods, which could limit its usage in certain cases.

However, while quantitative metrics provide insight into how label propagation performs, they do not fully capture the qualitative aspects of the segmentations, specially for low number of sparse points. In this cases, the visual aspects of the segmentation, like over-segmented

shapes, or leave areas unlabeled, become important for understanding how well the methods work in, particularly for experts to use them. These visual differences highlight key trade-offs that the metrics alone do not capture.

Qualitative evaluation When analyzing the segmentations with 30 sparse point-labels, SPX propagation often shows noticeable over-segmentation. Due to the use of low number of sparse points, the resulting superpixels are less but larger, each of them covering a big area. As each of the superpixels gets labeled, this leads to an excessive coverage of the image. In consequence, SAM2+SPX also shows this effect. Using only 30 sparse points SAM2 propagation usually generates smaller propagations than each of the superpixels, so SAM2 propagated masks are often completely covered in this approach. Figure 4.1 highlights how SPX-based segmentations tend to over-segment in both datasets. Although SPX achieves higher mPA score due to its broader coverage, this comes at the cost of scene understanding. The over-segmentation often hides important details and objects, making the segmentation harder to interpret and less useful in real-world situations.

Focusing on SAM2+DINOv2 method, it excels in scenarios with multiple similar objects belonging to the same class, such as fish. In these cases, it can identify and correctly label new, previously unlabeled objects, providing more complete segmentations. However, it sometimes creates incorrect new labels, especially in scenes without these features, where it is harder to distinguish between objects of different classes. Even with this drawback, SAM2+DINOv2's ability to label previously unlabeled areas adds value, especially when other methods miss these objects. Figure 4.2 shows the strengths of this approach.

4.2.2 Impact of Sparse Labels Location

As we have seen, the number of initial sparse point-labels affect the performance of all label augmentation methods (Table 4.1). However, their spatial distribution with respect to the objects in the image can significantly influence the label propagation process. Poorly positioned sparse labels such as those falling in not interesting or "background" areas, along object boundaries, or unevenly distributed across the image, may lead to inaccurate or incomplete augmentations (random or grid-based point distribution). An ideal alternative involves precise label placement guided by experts. Expert-labeled points are typically centered in the objects of interest, avoiding background areas and minimizing propagation errors caused by ambiguous boundary locations.

To simulate the benefits of expert labeling in our analysis, we developed a *smart* sparse labeling strategy that can be used on datasets that have dense ground truth semantic labels. The algorithm uses SAM2 segmentation model to identify and segment objects in the image, calculates the centroids of each segment, and ensures that selected points avoid background areas by cross-referencing the ground truth semantic labels.

Table 4.3 compares mPA and mIoU for each label augmentation approach using random and *smart* sparse point-labels. The mPA and mIoU metrics show that performance with random points is generally better, although SAM2 and SAM2+DINOv2 come close in mIoU, while SPX-based methods achieve higher mPA using *smart* points. This lower performance could be because the smart approach does not generate evenly distributed points sometimes, selecting points in regions of the same object, that then will likely be expanded into small segments, leaving unlabeled pixels, and therefore affecting the metrics. This effect is amplified when using few point-labels. However, similarly than previous experiments, we can see how these commonly used semantic segmentation metrics do not always fully capture how

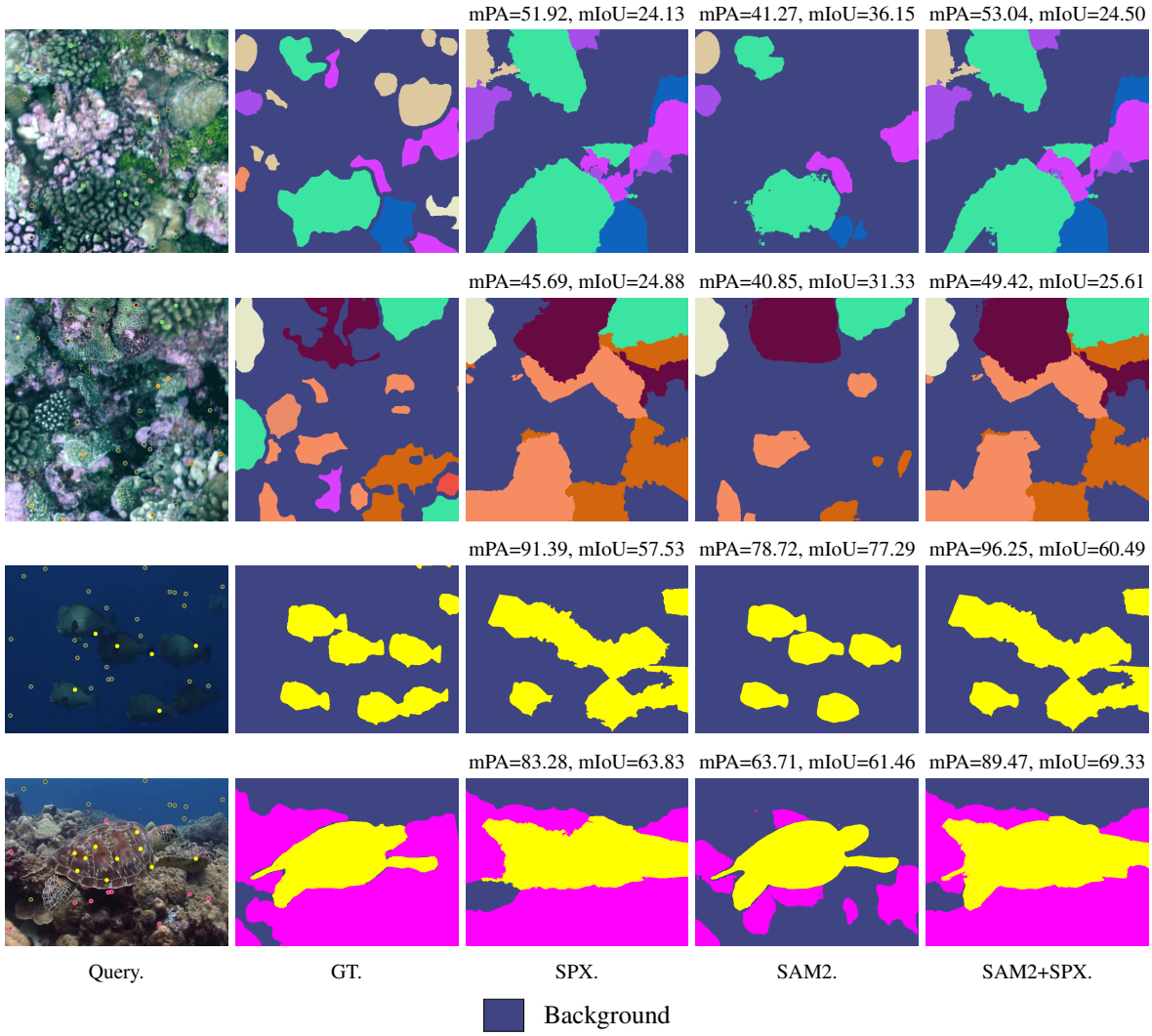


Figure 4.1: Over-segmentation examples on UCSD Mosaics (first two rows) and SUIM (last two rows) datasets with 30 sparse points by SPX-based label augmentation methods. SPX over-segmentation benefits mPA by correctly labeling many pixels, even though it also includes incorrectly labeled pixels in the segmentation. SAM2 obtains almost always better mIoU values than SPX and SAM2+SPX. In last row, SAM2 segments the turtle perfectly, providing a clearer and more interpretable result for this scene, despite having lower mPA and mIoU due to its incomplete segmentation of the seabed.

well the methods enable scene interpretation. In the qualitative example, we observe that SPX-based methods fail to leave any region as background, over-segmenting the entire image. This happens because no sparse points are labeled as background, and the method can only assign superpixel labels based on the sparse point-labels provided. Conversely, SAM2 and SAM2+DINOv2 despite not having much better mPA and mIoU, the generate much cleaner segmentation results,

This strategy is a preliminary version and requires further development to address its limitations. Currently, it does not always ensure an even distribution of points across the image and, because of this, may leave some important objects without any labeled points, which was the primary objective of the approach. Future refinements will focus on optimizing point selection to improve coverage across all relevant objects while reducing redundancy, ensuring better representation for a more accurate label augmentation.

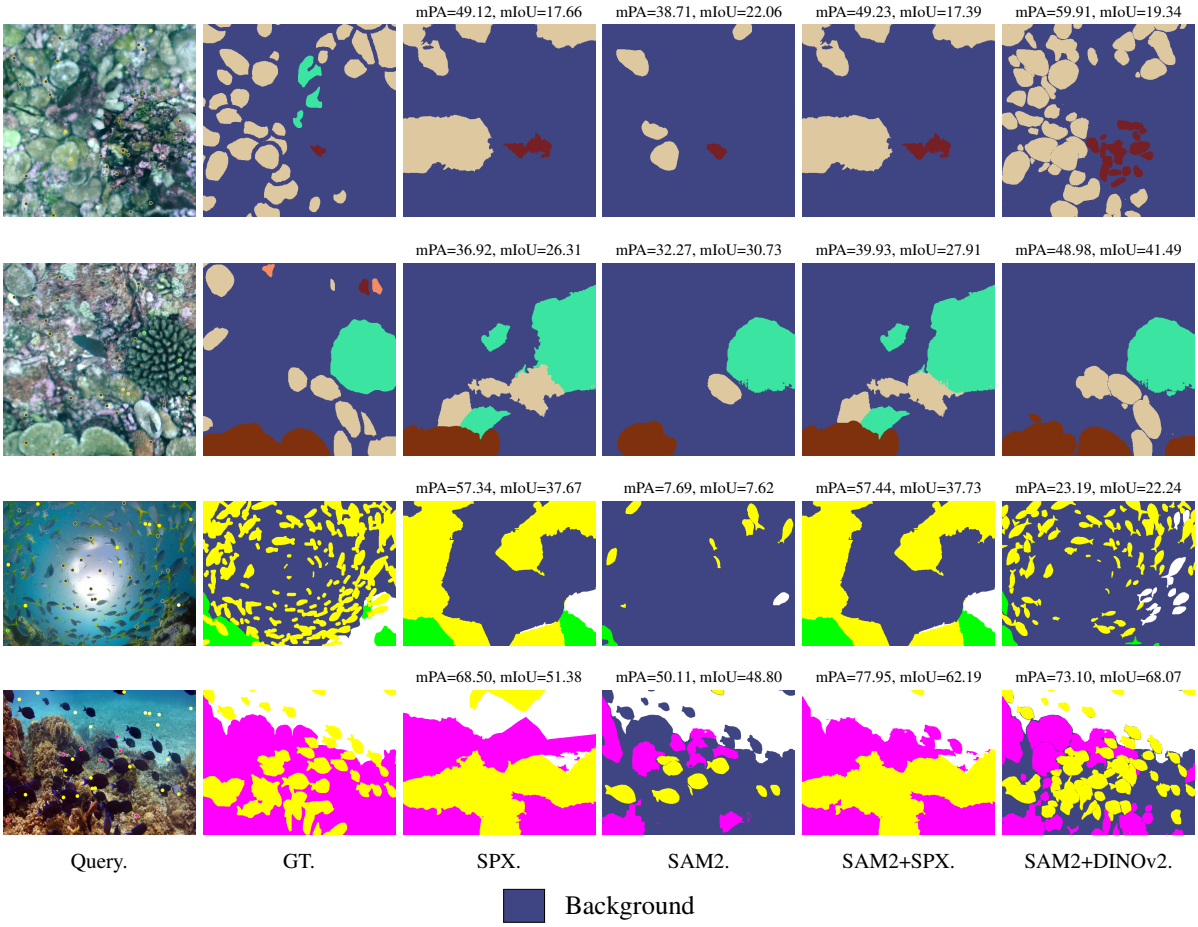


Figure 4.2: Examples showing good qualitative performance of SAM2+DINOv2 across four scenarios from UCSD Mosaics (first two rows) and SUIM (last two rows) datasets, compared to SAM2 and SPX-based methods. Despite mPA and mIoU values are lower in some cases (rows 1 and 3), SAM2+DINOv2 demonstrates the ability to find and correctly label new objects that were previously unlabeled by SAM2 method, particularly in scenarios with multiple instances of the same class. While some incorrectly labeled objects are also present, these examples highlight how metric values may not always capture label augmentation quality in terms of scene understanding.

4.2.3 Training with Augmented Labels

As introduced in Section 3.3, an application of the previous label augmentation methods is when we want to train a semantic segmentation model and we require a great amount of dense semantic segmentations. In this context, we could either manually annotate all these images or manually label some sparse points and apply our best label augmentation approach to generate pseudo-labels. In this section, we evaluate the effectiveness of these pseudo-labels by training a model using SegFormer [23] and analyzing its performance.

In this context, we use SegFormer-B5, which is a relatively small model (82M parameters) oriented to address semantic segmentation task and which enables us to efficiently run multiple experiments only varying the labels source of the training set. The remaining training and test configuration is fixed for all the experiments in order to ensure a fair comparison. Specifically, the training configuration includes the Adam optimizer with a learning rate of 1e-4, for 5 epochs. The training split consists of 80% of the 3974 train images in the UCSD Mosaic dataset, with the remaining 20% reserved for validation, chosen randomly.

Table 4.4 compares the performance of SegFormer trained with labels from different sources and varying numbers of labeled points. Thus, we first include as baseline the SegFormer per-

Table 4.3: Comparison of mPA and mIoU of label augmentations from 30 sparse point-labels distributed randomly or with the *smart* approach. **Bold** and underlined means highest and second highest, respectively. The images show a qualitative example of label augmentation using *smart* distribution where is noticeable that the position of each sparse point is centered in the different objects. However it may select points in regions that belong to the same object, as happens in textures with high relief, such as corals. SPX-based approaches incorrectly label all background points, leading into confusing segmentations, while SAM2 and SAM2+DINIOv2 obtain cleaner results.

Method	Sparse Points Distribution	mPA	mIoU
SPX	random	67.86	45.55
	smart	<u>77.74</u>	38.87
SAM2	random	52.11	49.30
	smart	48.65	46.09
SAM2+SPX	random	73.28	<u>49.91</u>
	smart	**83.33**	42.95
SAM2+DINOv2	random	61.31	**50.00**
	smart	54.79	47.12

Query. GT. SPX. SAM2. SAM2+SPX. SAM2+DINOv2

formance when trained with the dense ground truth masks of UCSD Mosaics. Besides, we compare the SegFormer performance when trained with the pseudo-labels generated from all the different label augmentation methods analyzed in the previous sections. All models are evaluated on the test split of the UCSD Mosaics dataset, using mPA and mIoU.

It is important to notice that, assuming that the ground truth labels are perfect, we can assert that the baseline performance is the best performance that we could get with the SegFormer model and the chosen training configuration. The results show that training with our pseudo-labels achieve competitive performance compared to the baseline, if we consider the difference of labeled points used. For 300 ($\sim$ 0.1% of image pixels) labeled points, SAM2+SPX achieve best mPA and mIoU. In contrast, when only 30 ($\sim$ 0.01% of image pixels) labeled points are used, SAM2+DINOv2 provides higher mIoU than SAM2+SPX.

We may conclude that with our label propagation approach we can considerably reduce the human labelling effort to train and get an accurate semantic segmentation model.

Table 4.4: Performance of SegFormer trained with different supervision strategies on the training set (3974 images) of UCSD Mosaics dataset. First row represents the model trained with dense ground truth labels as a baseline, using all labeled points of every image. Rows 2 to 5 show the results for training with pseudo-labels generated from different label augmentation approaches, varying the number of initial sparse point-labels (30 and 300).

Training strategy			Test results	
Label Source	Supervision type	Num labeled points	mPA ↑	mIoU ↑
Dense ground truth	dense	262,144 (all)	67.60	56.91
SAM2 + SPX	sparse	300	**63.74**	**47.38**
SAM2 + DINOv2	sparse	300	62.01	43.59
SAM2 + SPX	sparse	30	**46.91**	31.99
SAM2 + DINOv2	sparse	30	42.48	**33.45**

Chapter 5

Conclusions, challenges and future work

5.1 Conclusions

We have developed a framework to explore label augmentation from sparse point-labels, leveraging the capabilities of foundation models in wildlife scenarios, where dense annotations are often unavailable. This framework addresses the common challenges of scarce labeling by introducing and evaluating new methods for propagating and refining sparse point-labels.

We have demonstrated that propagating sparse point-labels with SAM2 generates more accurate segmentations, particularly in terms of preserving object shapes, compared to current state-of-the-art superpixel-based methods. However, with very few sparse labels, our SAM2 propagation method tends to leave significant portions of the image unlabeled. To address this limitation, we proposed two refinement approaches to label these remaining areas. The first combines SAM2 propagation with a superpixel-based label propagation method [35] (SAM2+SPX), while the second utilizes SAM2's ability to generate masks without prompts and assigns labels to these new masks by identifying the most similar labeled mask in the DINOv2 embedding space (SAM2+DINOv2).

Our experiments highlighted the strengths and trade-offs of these approaches. SAM2+SPX emerged as the most balanced method, offering strong performance in mPA, mIoU, and computational efficiency. Nevertheless, qualitative analysis showed that these standard semantic segmentation metrics do not always capture segmentation quality, particularly for object boundaries and fine details. SAM2+DINOv2 showed really good qualitative results in certain scenarios where the number of instances of objects of the same class is high, finding and label correctly most of them. We also investigated the impact of label quantity and spatial distribution by simulating expert labeling, demonstrating that not just the number, but also the placement of sparse labels significantly affects the results. Finally, we showed that the augmented labels can effectively be used to train SegFormer, a deep learning semantic segmentation model, achieving competitive performance compared to it trained with dense annotations, even when using a tiny fraction of labeled points.

Overall, this work demonstrates the potential of leveraging foundation models to obtain dense semantic segmentations from a small fraction of labeled image pixels, providing valuable insights for researchers in other disciplines to better understand these scenes, and offering a weak supervision approach for deep learning semantic segmentation models, achieving competitive performance.

5.2 Challenges and limitations

The challenges and limitations encountered during this work highlight important areas for improvement. The limitations with label propagation are closely related to the sparse points themselves. While the number of labeled points is important, their location also plays a meaningful role. If no point is placed within a specific class, objects of that class cannot be segmented correctly. Similarly, points positioned near the boundaries of objects can lead to inaccurate segmentations, as they may cause incorrect propagation across different regions.

Regarding refinement methods, SAM2+SPX performs well when the number of sparse points is moderate (e.g., 300), but it tends to over-segment objects, particularly when very few labels (e.g., 30) are available. In such cases, less superpixels are generated, each covering broader areas, often overriding accurate propagation achieved by SAM2. On the other hand, SAM2+DINOv2 excels in scenarios with multiple objects of the same class, but it often mislabels objects due to high feature similarity between different classes. This is specially challenging when no points are available for certain classes. Additionally, SAM2+DINOv2 is computationally expensive, which could restrict its use in certain tasks.

5.3 Future work

This work opens several directions for future research. First, we aim to develop novel metrics that more accurately represent the quality of our segmentations, beyond traditional metrics like mPA and mIoU. We also could enhance SAM2+DINOv2 method by not only finding similar objects within an image, but also within the whole dataset, enabling accurate labeling even when a class is absent in the current image's sparse points.

Another possible direction is to further explore *smart* sparse point distribution, optimizing for even class coverage. Pushing this idea, we could develop an interactive tool that guides the human experts from other disciplines in placing sparse points strategically, maximizing class coverage and reducing the human effort wasted for labelling.

Finally, we could explore the use of diverse label types, such as bounding boxes or even text prompts, leveraging large language models capabilities, for more robust and flexible label augmentation.

Bibliography

[1] Fei Chai, Kenneth S Johnson, Hervé Claustre, Xiaogang Xing, Yuntao Wang, Emmanuel Boss, Stephen Riser, Katja Fennel, Oscar Schofield, and Adrienne Sutton. Monitoring ocean biogeochemistry with autonomous platforms. *Nature Reviews Earth & Environment*, 1(6):315–326, 2020.

[2] Marc Besson, Jamie Alison, Kim Bjerge, Thomas E Gorochowski, Toke T Høye, Tommaso Jucker, Hjalte MR Mann, and Christopher F Clements. Towards the fully automated monitoring of ecological communities. *Ecology Letters*, 25(12):2753–2775, 2022.

[3] Oliver M Cliff, Debra L Saunders, and Robert Fitch. Robotic ecology: Tracking small dynamic animals with an autonomous aerial vehicle. *Science Robotics*, 3(23):eaat8409, 2018.

[4] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[5] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[7] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[8] Cesar Borja and Ana C. Murillo. Comprensión automática de escenas en imágenes de entornos submarinos. *Revista Iberoamericana de Automática e Informática industrial*, 21(4):374–382, may 2024.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[10] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[11] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[12] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[14] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[15] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[16] Michael Van den Bergh, Xavier Boix, Gemma Roig, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. *International Journal of Computer Vision*, 111:298–314, 2015.

[17] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV 10*, pages 705–718. Springer, 2008.

[18] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[20] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[21] Liang-Chieh Chen. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[22] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[23] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[25] Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5987–5996, 2018.

[26] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1049–1058, 2016.

[27] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. *Computer Vision–ECCV 2016*, pages 549–565, 2016.

[28] Junsong Fan and Zhaoxiang Zhang. Toward practical weakly supervised semantic segmentation via point-level supervision. *International Journal of Computer Vision*, 131(12):3252–3271, 2023.

[29] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4981–4990. IEEE, 2018.

[30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[31] Nasim Souly, Concetto Spampinato, and Mubarak Shah. Semi supervised semantic segmentation using generative adversarial network. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017.

[32] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13706–13715, 2020.

[33] Yachao Zhang, Zonghao Li, Yuan Xie, Yanyun Qu, Cuihua Li, and Tao Mei. Weakly supervised semantic segmentation for large-scale point cloud. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3421–3429, 2021.

[34] Inigo Alonso, Matan Yuval, Gal Eyal, Tali Treibitz, and Ana C Murillo. Coralseg: Learning coral segmentation from sparse annotations. *Journal of Field Robotics*, 36(8):1456–1477, 2019.

[35] Scarlett Raine, Ross Marchant, Brano Kusy, Frederic Maire, and Tobias Fischer. Point label aware superpixels for multi-species segmentation of underwater imagery. *IEEE Robotics and Automation Letters*, 7(3):8291–8298, 2022.

[36] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 352–368, 2018.

[37] Scarlett Raine, Ross Marchant, Brano Kusy, Frederic Maire, Niko Sunderhauf, and Tobias Fischer. Human-in-the-loop segmentation of multi-species coral imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2723–2732, 2024.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[39] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[40] Clinton B Edwards, Yoan Eynaud, Gareth J Williams, Nicole E Pedersen, Brian J Zgliczynski, Arthur CR Gleason, Jennifer E Smith, and Stuart A Sandin. Large-area imaging reveals biologically driven non-random spatial patterns of corals at a remote reef. *Coral Reefs*, 36:1291–1305, 2017.

[41] Md Jahidul Islam, Chelsey Edge, Yuyang Xiao, Peigen Luo, Muntaqim Mehtaz, Christopher Morse, Sadman Sakib Enan, and Junaed Sattar. Semantic segmentation of underwater imagery: Dataset and benchmark. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1769–1776. IEEE, 2020.

Appendix A

Software and Algorithmic details

This appendix includes a brief description and pointer to the software tool developed as part of this thesis as well as a more detailed algorithmic description of the label assignment algorithm.

A.1 Software

The software implementing our proposed framework is publicly available at `https://github.com/cborjamoreno/PL_augmentation`. It offers an easy-to-use tool for researchers, even those without AI expertise, to augment sparse point-labels and create dense semantic segmentations for their specific needs.

A.2 Algorithms

Algorithm 1: Label Assignment. Pseudo-code of label assignment algorithm.

Input: $M = \{m_1, m_2, \ldots, m_n\}$, $P_L = \{(p_1, l_1), (p_2, l_2), \ldots, (p_n, l_n)\}$
Output: m_{out}
$m_{\text{out}} \leftarrow \emptyset$;
`overlap_dict_temp` $\leftarrow \emptyset$;
`overlap_dict` $\leftarrow \emptyset$;
`/* Detect overlapping masks */`
foreach $m_i \in M$ **do**
  foreach $m_j \in M$ *where* $i \neq j$ **do**
    $b_i, b_j \leftarrow$ bounding boxes of m_i, m_j;
    if b_i *and* b_j *overlap* **then**
      $P_{\text{overlap}} \leftarrow m_i \cap m_j$;
      `overlap_dict_temp`$[P_{\text{overlap}}] \leftarrow$ `overlap_dict_temp`$[P_{\text{overlap}}] + [i, j]$;
    end
  end
end
`/* Group pixels by overlapping masks */`
foreach $(P_{overlap}$,*involved*$) \in$ *overlap_dict_temp* **do**
  `overlap_dict`[`involved`] $\leftarrow$ append P_{overlap};
end
delete `overlap_dict_temp`
`/* Assign label to overlap areas */`
foreach $(involved, P_{overlap}) \in$ *overlap_dict* **do**
  $P_{L\text{overlap}} \leftarrow$ P_L within $P_{\texttt{overlap}}$ for each label;
  if *majority_label exists* **then**
    $M_{\text{out}}[P_{\text{overlap}}] \leftarrow$ `majority_label`;
  else
    $P_{L\text{involved}} \leftarrow P_L$ within $M_{involved}$;
    $c \leftarrow$ Centroid of $P_{\texttt{overlap}}$;
    $(p_{\text{closest}}, l_{\text{closest}}) \leftarrow \arg\min_{p,l \in P_{L\text{involved}}} \|c - p\|_2$;
    $M_{\text{out}}[P_{\texttt{overlap}}] \leftarrow l_{\text{closest}}$;
  end
end
foreach $M_i \in$ *Masks* **do**
  $P_{\text{non-overlap}} \leftarrow$ Identify non-overlapping pixels in M_i;
  $M_{\text{out}}[P_{\texttt{non-overlap}}] \leftarrow$ Label of $P_{\text{non-overlap}}$;
end
return M_{out};

Appendix B

Additional Results

B.1 Similarity threshold selection

To determine the threshold similarity in Feature-Guided Mask Matching approach (3.2.4), we manually select a value within the range [0.0, 1.0]. This threshold establishes whether the label of a labeled mask is considered during the assignment of a new label to an unlabeled mask. The evaluation of label augmentation was performed using threshold values across the range. For this experiment we used the 696 test images from the UCSD Mosaics dataset, so we do not use images that were evaluated in Label Augmentation results (4.2.1). 30 sparse point labels were used for the augmentation.

τ	mPA (%)	mIoU (%)
0.0	55.08	41.27
0.1	55.21	41.34
0.2	55.69	41.35
0.3	57.09	41.05
0.4	59.02	43.06
0.5	**59.99**	47.37
0.6	58.98	**50.08**
0.7	55.99	49.95
0.8	53.17	48.28
0.9	52.29	47.54
1.0	52.21	47.49

Table B.1: Performance of label augmentation for SAM2+DINOv2 approach across different threshold similarity (τ) values.

The best mPA metric is obtained with 0.5 $\tau = 0.5$, while the best mIoU is achieved with $\tau = 0.6$. As the difference between mIoU is greater, we chose to use $\tau = 0.6$ as the predefined threshold value.

B.2 Qualitative evaluation in other datasets

We evaluated qualitatively two additional set of images provided by The Byrness Lab, UMass: Boston. Both with sparse segmentations:

- **Miscellaneous Substrate Imagery** A set of 46 images, taken in the Gulf of Maine, each with a size of 1024×768. These images are randomly sampled and contain 20 semantic classes.
- **Kelp Imagery** A set of 113 kelp images, each 1024×768 in size and grid-sampled, containing 5 semantic classes.

Figures B.1 and B.2 show label augmentation examples with all methods for Miscellaneous Substrate Imagery and Kelp Imagery datasets, respectively.

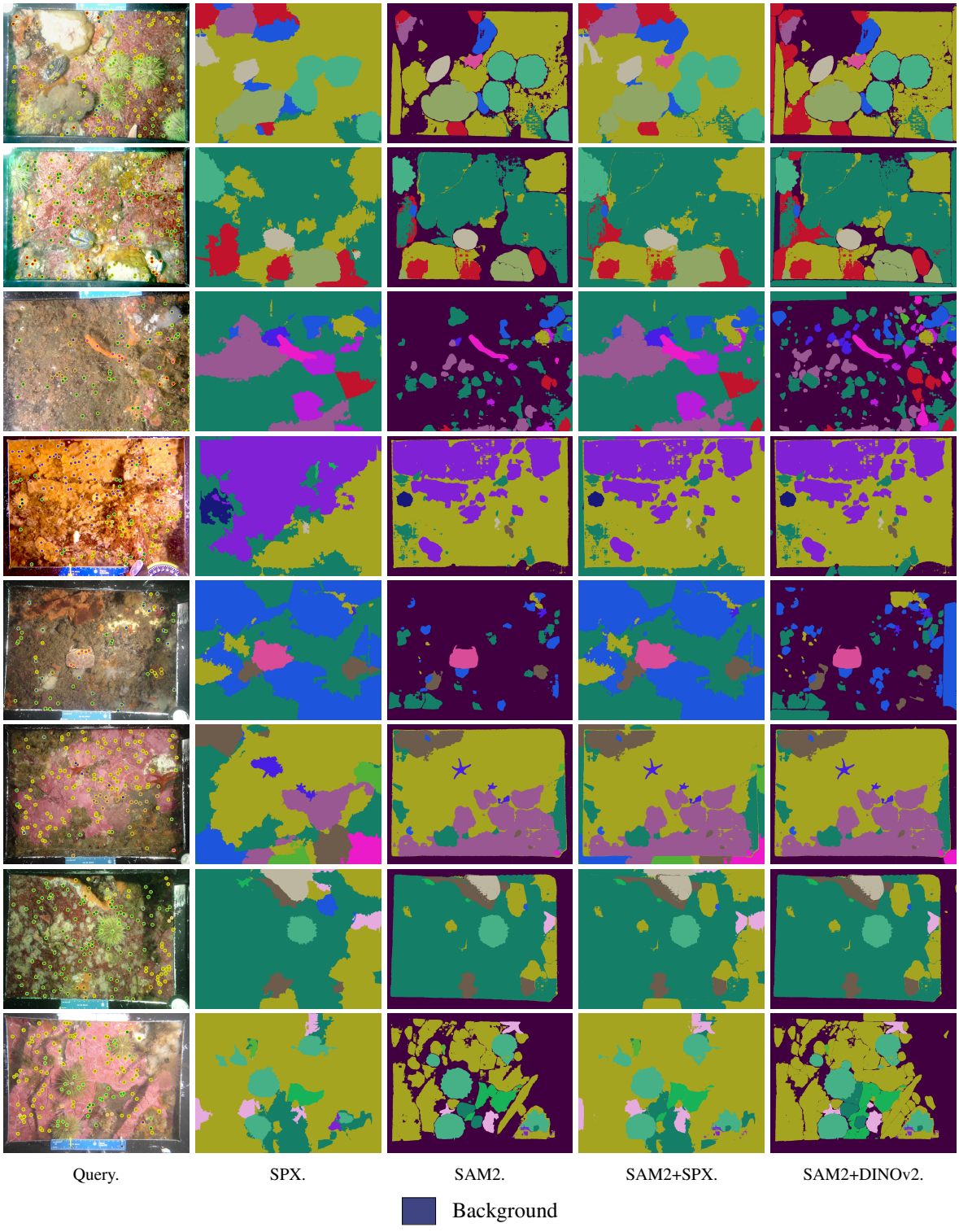


Figure B.1: Label augmentation examples in Miscellaneous Substrate Imagery dataset.

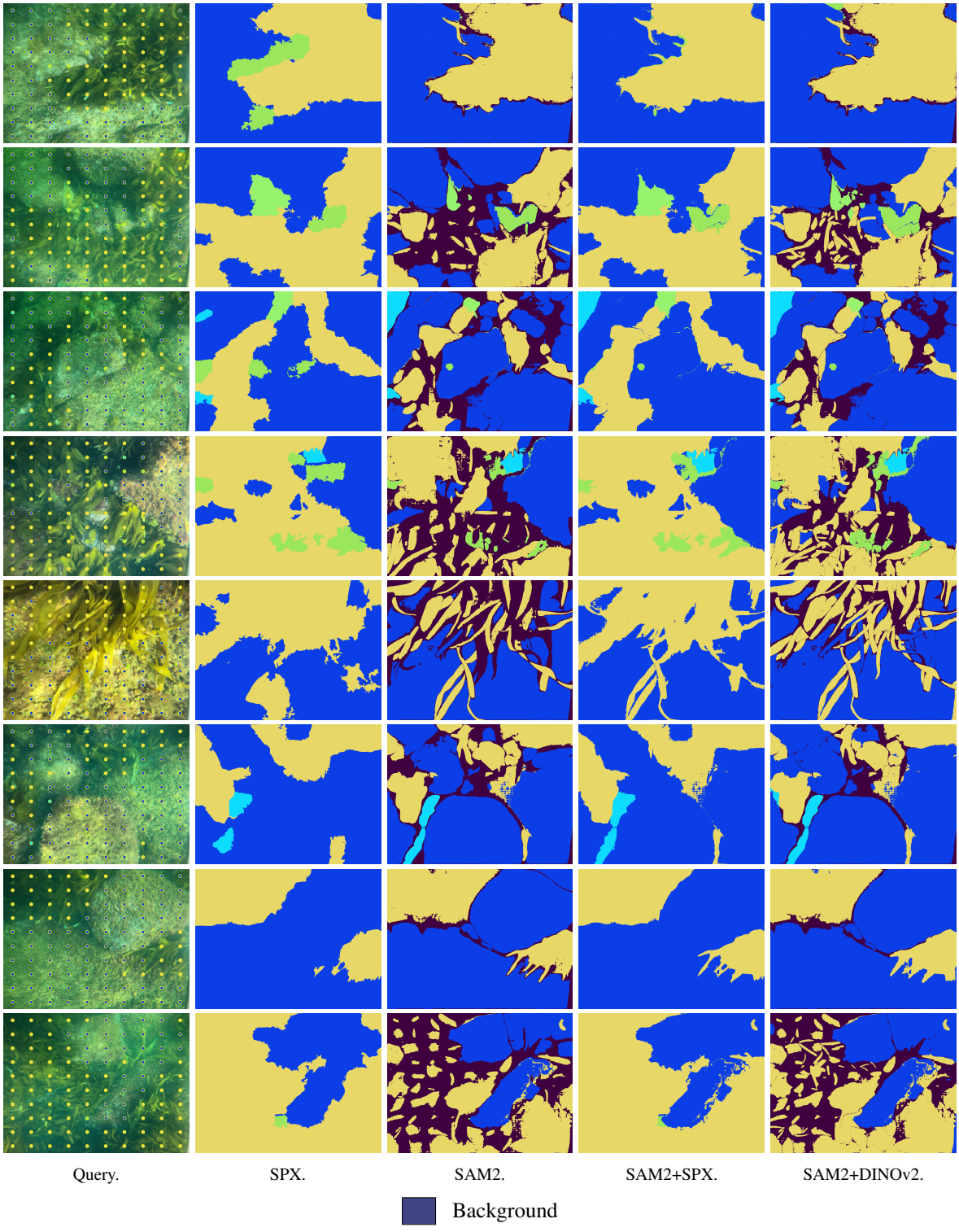


Figure B.2: Label augmentation examples in Kelp Imagery dataset.