Static Segmentation by Tracking: A Frustratingly Label-Efficient Approach to Fine-Grained Segmentation

Zhenyang Feng[1], Zihe Wang[1], Saul Ibaven Bueno[1], Tomasz Frelek[1], Advikaa Ramesh[1], Jingyan Bai[1], Lemeng Wang[1], Zanming Huang[1], Jianyang Gu[1], Jinsu Yoo[1], Tai-Yu Pan[1], Arpita Chowdhury[1], Michelle Ramirez[1], Elizabeth G. Campolongo[1], Matthew J. Thompson[1], Christopher G. Lawrence[2], Sydne Record[3], Neil Rosser[4], Anuj Karpatne[5], Daniel Rubenstein[2], Hilmar Lapp[6], Charles V. Stewart[7], Tanya Berger-Wolf[1], Yu Su[1], Wei-Lun Chao[1]

[1]The Ohio State University, [2]Princeton University, [3]University of Maine, [4]University of Miami, [5]Virginia Tech, [6]Duke University, [7]Rensselaer Polytechnic Institute

Abstract

*We study image segmentation in the biological domain, particularly trait and part segmentation from specimen images (*e.g.*, butterfly wing stripes or beetle body parts). This is a crucial, fine-grained task that aids in understanding the biology of organisms. The conventional approach involves hand-labeling masks, often for hundreds of images per species, and training a segmentation model to generalize these labels to other images, which can be exceedingly laborious. We present a label-efficient method named* ***Static Segmentation by Tracking (SST)****.* SST *is built upon the insight: while specimens of the same species have inherent variations, the traits and parts we aim to segment show up consistently. This motivates us to concatenate specimen images into a "pseudo-video" and reframe trait and part segmentation as a* tracking *problem. Concretely,* SST *generates masks for unlabeled images by* propagating *annotated or predicted masks from the "pseudo-preceding" images. Powered by Segment Anything Model 2 (SAM 2) initially developed for video segmentation, we show that* SST *can achieve high-quality trait and part segmentation with merely one labeled image per species—a breakthrough for analyzing specimen images. We further develop a cycle-consistent loss to fine-tune the model, again using one labeled image. Additionally, we highlight the broader potential of* SST*, including one-shot instance segmentation on images taken in the wild and trait-based image retrieval.*

1. Introduction

Understanding sources and patterns of intra-specific variation in traits (*e.g.*, morphological characteristics, such as fin length for a fish, wing size for a beetle) is a central goal of evolutionary and ecological study [11, 18]. Intra-specific trait variation provides a currency for assessing the roles of abiotic and biotic processes on community assembly, as it reflects the mechanisms driving species occurrence and responses to change [82]. Museum specimens present an untapped resource for curating information on intra-specific trait variation of species morphology. Up until now, it has been difficult to harvest trait information from museum specimens due to the sheer amount of manual labor needed to make such measurements. Automatic segmentation of morphological traits from specimen images has the potential to scale up the measurement of traits and free up researchers to focus on analysis and interpretation. This paper originated from an interdisciplinary collaboration between biologists and computer scientists aimed at segmenting images of organismal specimens to measure variation in traits to fill this much-needed knowledge gap.

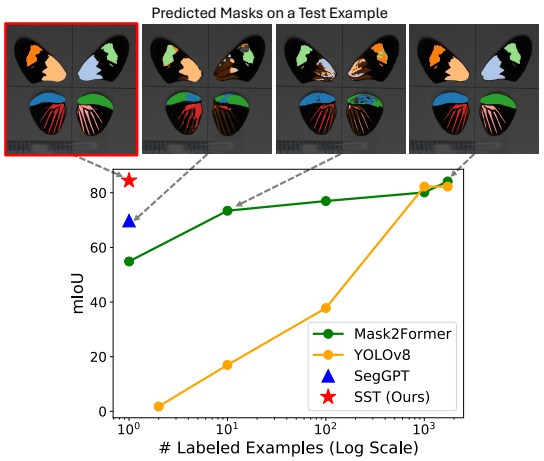

Figure 1. **Static Segmentation by Tracking (SST)** achieves high-quality trait segmentation (on *Heliconius erato lativitta*) with one labeled example, compared to other one/many-shot baselines.

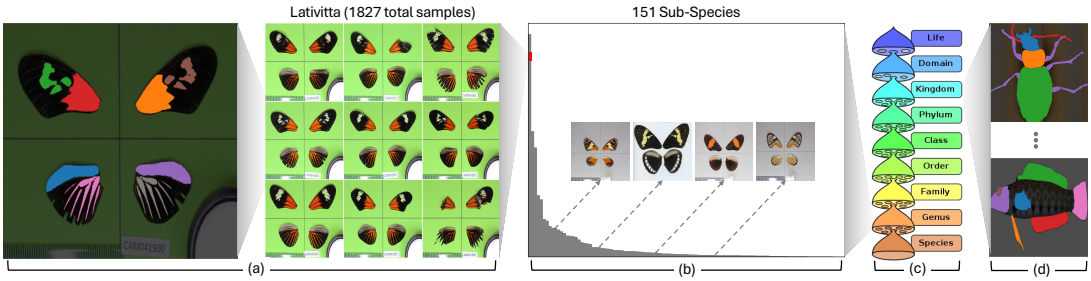

Figure 2. **Illustration of the Trait Segmentation Problem From Specimen Images.** (a) Specimen samples of *Heliconius erato lativitta*, and one example of segmentation masks. (b) The histogram of samples per *sub-species* in the Cambridge Butterfly Collection [36], with exemplar images. (c) These sub-species belong to a specific genus named *Heliconius*, which is under the suborder Rhopalocera that covers all the $> 10,000$ butterfly species worldwide. (d) Trait segmentation is important for other animals such as beetles and fishes.

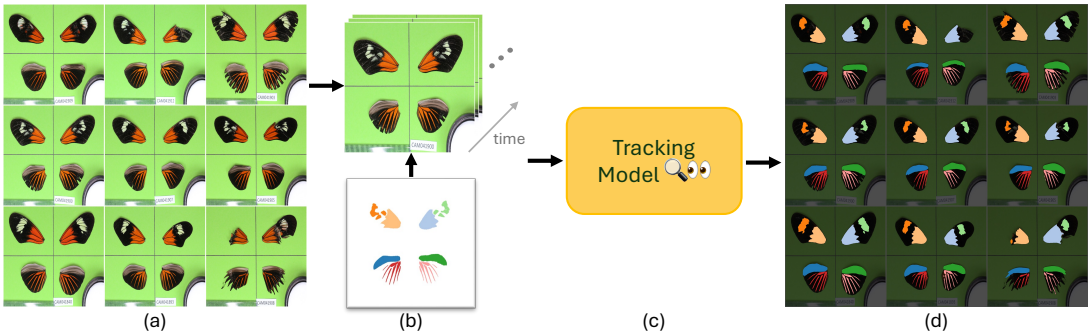

Figure 3. **Illustration of Our Approach Static Segmentation by Tracking (SST).** (a) Different specimens of the same species. (b) We concatenate these static, non-sequential images into a pseudo-video. (c) The annotated masks of the first image are treated as the prompt to a tracking algorithm such as SAM 2 [70]. (d) SST can generate high-quality trait segmentation in a one-shot fashion.

Training a segmentation model [12, 21, 37, 41, 52] is arguably the most straightforward approach to this problem. However, it requires humans to annotate traits on tens, if not hundreds, of images per species to ensure the model can generalize well to other images. This process is itself laborious, let alone there are millions of species on Earth and many of them do not have sufficient samples for labeling (see Fig. 2). Several recent segmentation algorithms focused on a few-shot setting, aiming to tailor the model to the concept of interest with a handful of labels [25, 81, 86, 87]. Nevertheless, most of them were designed for a single new concept at once (*e.g.*, the whole beetle instance) rather than multiple concepts jointly (*e.g.*, beetle head, antennae, and elytra case). Even for a single trait, they usually fail to capture the fine shape, performing much worse than models trained with many samples (see Sec. 4). We thus ask,

How can we perform fine-grained segmentation on specimen images without a large amount of labeled data?

We begin with a deeper look at specimen images, especially those of the same species. We have several key observations (see Fig. 2). From a *macro* view that sees a specimen as a "whole," since specimens are made from biological instances with inherent variations, they are doomed to look differently even from the same species; some even have damaged parts. However, from a *micro* view that sees a specimen as a "composition" of traits and parts—*the components we aim to segment*—specimens of the same species look quite similar in terms of their trait and part layouts. Unless damaged, these traits and parts consistently show up and have controlled spatial relationships with each other. Importantly, each has distinct characteristics, such as colors, shapes, patterns, and relative positions, offering rich cues to identify and locate it across specimens of the same species.

Taking these insights, we propose to reframe trait and part segmentation of specimen images as a *tracking* problem. Tracking is the task of taking an initial set of instances, creating a unique ID for each of them, and then tracking them as they move around frames in a video [43, 92]. *In our context, the instances are distinct traits and parts, each marked by a mask and a unique label.* While we do not have a video but a set of static specimen images, the variations of each trait or part across images—such as mild differences in sizes, locations, orientations, shapes, and colors—are frequently seen across video frames as a result of camera poses, motion, deformation, and lighting conditions. One may even view damaged parts as occlusions. This motivates

us to concatenate *static, non-sequential* specimen images into a "pseudo-video" and apply a tracking algorithm to locate and segment individual traits and parts across frames, *given only the annotated masks of the first frame (Fig. 3).*

We name our approach **Static Segmentation by Tracking (SST)**, which *lifts an image segmentation problem into a video tracking problem, using the characteristic of the latter to approach the former in a frustratingly label-efficient, one-shot way!* In essence, what the model is tasked to do is to locate each annotated mask (of the first frame) in the succeeding frames, and then *propagate* and *deform* the mask from the first frame to the succeeding frames.

We implement SST using the recently released Segment Anything Model 2 (SAM 2) [70], which was developed for video segmentation. Given the annotated masks of the first frame as the prompt, SAM 2 is capable of segmenting them over frames. We evaluate SST on three specimen image datasets, Cambridge Butterfly [36], NEON Beetle [22], and Fish-Vista [50]. SST demonstrates significantly better performance than the other one-shot baselines such as SegGPT [87] in trait and part segmentation. Surprisingly, in several scenarios, SST even surpasses segmentation models trained with ample labeled data, such as Mask2Former [15] and YOLOv8 [33] (see Fig. 1). *We attribute this to the fact that* SST *does not treat labeled and unlabeled images as IID samples—an assumption underlying most of the image segmentation algorithms—but explicitly leverages their dependency to facilitate segmentation.* We humbly see this as a breakthrough for analyzing specimen images.

When does SST fail? Seeing SST's remarkable one-shot segmentation performance of traits and parts, we conduct a pressure test, aiming to understand under what circumstances SST may fail. We add scaling, translations, and rotations to the original specimen images. We find that SST is quite robust to these variations if within a mild degree, but could degrade drastically under huge variations. We argue that if the specimens were carefully made and the images were taken in a canonical camera pose, then SST should be easily plugged and played to analyze specimen images.

Can we further improve SST? SST is built upon a pre-trained video segmentation model. We surmise that one reason for the degradation is the lack of huge variations in the pre-training video data, essentially out-of-distribution. While one may overcome this by fine-tuning the model with artificially augmented variations, it is hard to anticipate all the variations that may appear in reality. We thus ask,

How can we adapt the model
once we receive the unlabeled, test image?

Our answer is an **Opening-Closing Cycle-Consistent Loss (OC-CCL)**. Given a set of test images $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ for segmentation, we append the one-shot labeled image $(\boldsymbol{x}_0, \boldsymbol{y}_0)$ as the *opening* frame and the *closing* frame, denoted by $\boldsymbol{x}_0$ and $\boldsymbol{x}_{N+1}$. At $\boldsymbol{x}_0$, the segmentation label $\boldsymbol{y}_0$ serves as the prompt to SAM 2, while at $\boldsymbol{x}_{N+1}$, we ask the model to first predict $\hat{\boldsymbol{y}}_{N+1}$ and use $\boldsymbol{y}_0$ as supervision to fine-tune the model. The rationale is that if the model wants to successfully segment the labeled image appended at the end, it must track traits and parts in the intermediate test images well. Our empirical results demonstrate the notable gain via OC-CCL, using a single labeled image to fine-tune the model at test time! One may view our approach as an instance of test-time training [42, 78, 85].

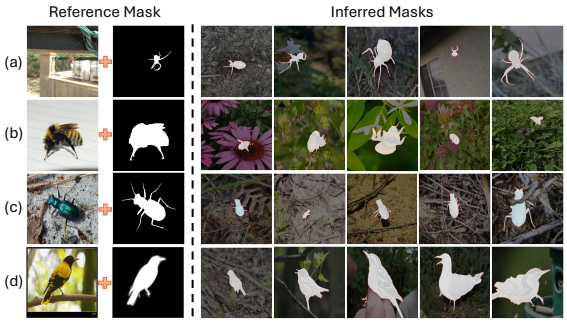

Figure 4. **Applying SST to Segmentation in the Wild.** (a-c) Spiders, bees, and beetles in iNaturalist [26]. (d) Birds in CUB [84].

Beyond specimen image segmentation. We explore other application scenarios of SST. First, we go beyond specimen images taken in the laboratory to consider (object-centric) animal photos taken in the wild. We find that using SST to segment traits and parts in these images is not trivial, as they may be heavily occluded by complex backgrounds, 3D pose variation, and 3D body deformation. However, if we take a level up, considering **animal instance segmentation**, SST performs surprisingly well, even if the "pseudo-video" is highly non-smooth with rapidly and arbitrarily changing backgrounds. For example, on the CUB dataset [84] for bird segmentation, **SST + OC-CCL** is competitive with the state-of-the-art algorithm dedicated to one-shot segmentation [25, 81, 87]. We view this as an emergent property of SAM 2: a foundation model trained for video segmentation can track and segment taxonomically related species over non-sequential, independently taken photographs.

Second, we investigate **trait-based image retrieval**. Unlike standard image retrieval which uses holistic image features to search for similar images [13, 19], we aim to find images that share a similar trait with the query image. We realize this idea by repurposing OC-CCL. Concretely, we claim that an image $\boldsymbol{x}'$ and the query image $\boldsymbol{x}$ share a certain trait if the trait mask $\boldsymbol{y}$ in $\boldsymbol{x}$ can be propagated to $\boldsymbol{x}'$ and then back to $\boldsymbol{x}$. On the Cambridge Butterfly dataset [36] of specimen images, we show that SST can precisely retrieve images containing a specific trait, like white bands on the forewings and hindwings of the butterfly.

Contribution. Our contributions are four-fold.

- We propose **Static Segmentation by Tracking (SST)**, a frustratingly label-efficient approach to fine-grained segmentation on specimen images. SST concatenates static, non-sequential images into a pseudo-video and applies tracking algorithms to segment traits. It needs merely one labeled image per species to complete the task, without any secret tweaking, making it readily *plug-and-play*!
- We propose **Opening-Closing Cycle-Consistent Loss (OC-CCL)** for one-shot fine-tuning to improve SST.
- We hand-labeled and semi-automatically labeled more than 836 and $2,831$ fine-grained trait masks on over 150 sub-species of butterfly specimen images, respectively. We also hand-labeled 180 beetle specimen images. We expect these labeled images to serve as the testbed for future research in specimen image segmentation.
- We demonstrate SST's potential in broader application scenarios, including instance segmentation on images taken in the wild and trait-based image retrieval.

Remark. Our main use case is specimen images. However, it by no means implies that our scope and applicability are "limited." First, specimens are one major way for biologists to understand organisms, and a gigantic amount of specimens have not yet been digitized and analyzed. For instance, there are an estimated 350,000,000 plant specimens deposited in the world's 3,400 herbaria [77]. This plethora of specimens represents just one of many taxonomic groups in the Tree of Life. Second, Earth has millions of species and each has distinctive sets of traits, demanding a versatile segmentation algorithm that can adapt to the idiosyncrasies of each species in a few-shot fashion. Third, while at first glance, the object-centric nature and the plain background may create a superficial impression that specimen images are much easier to deal with than natural images (*e.g.*, MS-COCO images [38]), our experiments show that segmenting fine-grained traits and parts from them is non-trivial, especially in a few-shot setting. Fourth, outside the computer vision community, object-centric images with canonical poses are a mainstream image source in various scientific domains. Take medical image processing as an example: many tasks are about MRI and CT-scan images and they are mostly taken in canonical poses. To sum up, our paper makes contributions to not only the vision community (*e.g.*, promoting a rarely studied but challenging task, providing data for benchmarking, and proposing a new way of thinking) but also other scientific communities (*e.g.*, making the measurement of traits much easier).

2. Related Work and Background

Image segmentation. Image segmentation has been a long-standing problem in computer vision, with various applications spanning across different fields [12, 21, 37, 41, 52]. Semantic and instance segmentation are arguably the most popular segmentation tasks nowadays, aiming to cluster pixels bearing the same semantic meanings and instances [23, 52]. While much of the focus has been on common, coarse-grained objects, several works have begun to explore more fine-grained part-level segmentation within common objects [29, 68]. In this paper, we aim to address a brand-new, extremely fine-grained segmentation problem.

Few-shot segmentation (FSS). Many state-of-the-art (SOTA) models have demonstrated impressive capabilities in image segmentation [15, 33]. However, one bottleneck to such a task is the laborious labeling efforts in creating pixel-level annotations for model training. To address such limitations, other works attempt to use few-shot learning techniques to provide high-quality segmentation masks to unseen classes, only with one or few image mask pairs as "support" set [24, 40, 81, 91, 93]. In this work, we propose a new few-shot learning approach and introduce a novel perspective to tackle the FSS problem.

Segmentation Anything Model (SAM). SAM [35] is a foundation model for image segmentation, achieving SOTA zero-shot segmentation performance. Previous works have shown its superiority in medical image segmentation [28, 44, 49], camouflaged object detection [79], semantic communication [80], and autonomous driving [76]. Recently, Segment Anything Model 2 (SAM 2) [70] was released, with extended capabilities on video segmentation tasks. Specifically, SAM 2 is designed to take in the prompts on any frame within a video sequence to help track and segment target objects and has been shown as the new SOTA model in video segmentation. We build upon SAM 2's superior video segmentation capability for fine-grained segmentation of non-sequential images in the biology domain.

Co-segmentation. The concept of co-segmentation was introduced by Rother et al. [71], who aimed to segment out the common foreground objects from multiple images. Most of the previous works focus on whole object instances and require some degrees of joint training [14], supervised clustering [34], or generation and ranking of region proposals [16, 83]. Our work can be viewed as a new approach to co-segmentation, leveraging the capabilities of video segmentation models to efficiently segment the common objects, parts, and traits across images given one labeled image.

3. Proposed Approach

Problem definition and notation. We study trait and part segmentation from specimens of the *same* species. Let $\boldsymbol{x} \in \mathbb{R}^{W \times H \times 3}$ denote a $W \times H$ image and $\boldsymbol{y} \in \{0,1\}^{W \times H \times C}$ denote the corresponding ground-truth masks of C distinct traits or parts. The goal is to develop a segmentation model f such that its output $\hat{\boldsymbol{y}} = f(\boldsymbol{x})$ matches $\boldsymbol{y}$.

Typically, one needs to collect a labeled training set with ample pairs of $(\boldsymbol{x}, \boldsymbol{y})$, and use it to train f in a supervised way. In this paper, we target the one-shot scenario, *i.e.*, building f using a single labeled image $(\boldsymbol{x}, \boldsymbol{y})$.

3.1. Static Segmentation by Tracking (SST)

At first glance, this looks like an impossible mission. However, the domain-specific properties described in Sec. 1 offer the rescue. Our proposed approach, **SST**, closely considers these properties and reframes trait and part segmentation as a tracking problem, which is inherently a one-shot task given a set of labeled instances in the first frame.

More specifically, let $\{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_N\}$ denote a *sequence* of video frames; a tracking algorithm aims to track the instances in $\boldsymbol{x}_0$, encoded by the label $\boldsymbol{y}_0$, across the remaining frames. In our context, we do not have a video but a *set* of N unlabeled images and one labeled image $(\boldsymbol{x}, \boldsymbol{y})$. Nevertheless, the domain-specific properties motivate us to create a "pseudo-video" by treating $\boldsymbol{x}$ as the first frame $\boldsymbol{x}_0$, followed by an arbitrary order of the rest. Please see Sec. 3.3 for a detailed discussion about the pseudo-video creation.

SAM 2 for SST. We adopt the recently released Segment Anything Model 2 (SAM 2) [70] for tracking, although other tracking algorithms may apply. We briefly introduce its model architecture and inference mechanism, followed by our usage. See also Sec. 2 for the background.

SAM 2 uses a promptable Transformer encoder-decoder f augmented with a memory bank $\boldsymbol{B}$ to process a video and generate masks (see Fig. 5). Let $\{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_N\}$ denote the video frames and $\{\boldsymbol{y}_0, \cdots, \boldsymbol{y}_N\}$ the corresponding ground-truth labels. When the label of $\boldsymbol{x}_n$ is not available, $\boldsymbol{y}_n = \varnothing$. Let us denote the predicted mask for $\boldsymbol{x}_n$ by $\hat{\boldsymbol{y}}_n$ and the updated memory bank by $\boldsymbol{B}_n$, which stores both the feature and mask information. $\boldsymbol{B}_n$ can then be accessed by the next frame $\boldsymbol{x}_{n+1}$ to connect consecutive frames. In the context of tracking, $\boldsymbol{B}_n$ can be interpreted as the updated state estimate after perceiving the measurement $\boldsymbol{x}_n$.

At each timestamp n, f takes the tuple $[\boldsymbol{x}_n, \boldsymbol{B}_{n-1}, \boldsymbol{y}_n]$ as input, where $\boldsymbol{y}_n$ is treated as the (optional) prompt. It then outputs the tuple $[\boldsymbol{B}_n, \hat{\boldsymbol{y}}_n]$, where $\boldsymbol{B}_n$ will be used as an input at the next timestamp,

$$[\boldsymbol{B}_n, \hat{\boldsymbol{y}}_n] = f\left([\boldsymbol{x}_n, \boldsymbol{B}_{n-1}, \boldsymbol{y}_n]\right). \tag{1}$$

In our context, we have $\boldsymbol{y}_n = \varnothing, \forall n > 0$. Namely, only the first frame $\boldsymbol{x}_0$ is labeled. Inputting $\boldsymbol{y}_0$ to f at timestamp 0 instructs the model on what to *segment*—the distinct traits and parts and their extents. The resulting memory bank $\boldsymbol{B}_0$ then carries such information to the successive frames, instructing the model on what to *track* over frames to generate the masks $\{\hat{\boldsymbol{y}}_1, \cdots, \hat{\boldsymbol{y}}_N\}$. See Fig. 3 for an illustration.

Remark. According to the original paper [70], SAM 2 is readily applicable to a batch of static, non-sequential images, *by setting the memory bank $\boldsymbol{B}_n$ to empty*. In essence, without the memory bank, SAM 2 simply treats every input image as IID samples and processes them independently.

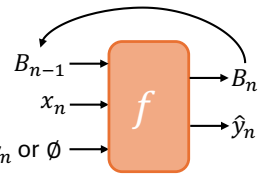

Figure 5. **Illustration of SAM 2's Mechanism [70].** It can be seen as a function f that takes in the encoded context from the previous frames B_{n-1}, the target image x_n, and an optional prompt y_n, and then outputs the encoded context including the current frame B_n and the predicted segmentation masks $\hat{y}_n$.

Our insight is that even if the input images are taken in a non-sequential fashion, whenever there exists useful dependency among them (*e.g.*, from the same species), SAM 2 has the potential to capture it in the subsequent frames. One just needs to let the memory bank update, not resetting it.

3.2. Opening-Closing Cycle-Consistent Loss

SST uses SAM 2 in a plug-and-play fashion without changing its pre-trained weights, even though our use case is beyond the training data distribution. It is thus expected that SST might fail when the transitions across static images are significantly out-of-distribution (OOD).

One intuitive way to address this is to fine-tune SAM 2. However, given merely one labeled image $(\boldsymbol{x}_0, \boldsymbol{y}_0)$, fine-tuning risks overfitting. Noticing that we have used $\boldsymbol{y}_0$ to prompt SAM 2, we face another challenge: *it is unclear how to use it "dually" as the label to supervise fine-tuning.*

We propose a novel approach, which leverages the flexibility of creating pseudo-videos. We can not only concatenate static images in a flexible order but also duplicate them and inject them at different timestamps to obtain multiple predictions for the same image. Specifically, we duplicate the labeled image $\boldsymbol{x}_0$ and append it at the end of the pseudo-video. The resulting video becomes $\{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_N, \boldsymbol{x}_{N+1}\}$, where $\boldsymbol{x}_{N+1} = \boldsymbol{x}_0$. Unlike timestamp 0 where $\boldsymbol{x}_0$ is used as the "opening frame" and $\boldsymbol{y}_0$ is inputted as the prompt, at timestamp $N+1$, $\boldsymbol{x}_{N+1}$ is treated as an unlabeled "closing frame" without prompts (*i.e.*, $\boldsymbol{y}_{N+1} = \varnothing$),

$$[\boldsymbol{B}_{N+1}, \hat{\boldsymbol{y}}_{N+1}] = f\left([\boldsymbol{x}_{N+1}, \boldsymbol{B}_N, \varnothing]\right). \tag{2}$$

Such an arrangement allows us to use $\boldsymbol{y}_0$, the ground-truth label of $\boldsymbol{x}_{N+1}$, to supervise the fine-tuning of f by minimizing the difference between $\hat{\boldsymbol{y}}_{N+1}$ and $\boldsymbol{y}_0$. The rationale is if f fails in intermediate frames, $\boldsymbol{B}_N$ will not carry useful information for segmenting the traits or parts in $\boldsymbol{x}_{N+1}$.

We employ a combination of the binary cross entropy (BCE) loss and the Dice loss for fine-tuning, which are commonly used in training a segmentation model. We name our fine-tuning objective function **Opening-Closing Cycle-**

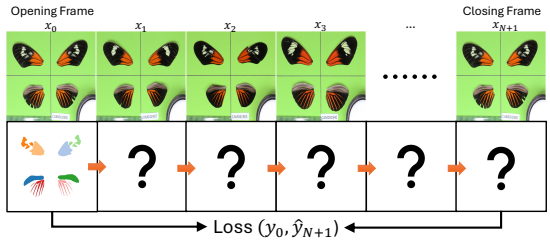

Figure 6. **Illustration of the Opening-Closing Cycle-Consistent Loss (OC-CCL).** By appending the labeled example as the last frame, we can evaluate the segmentation quality by comparing the predicted masks with the ground truth.

Consistent Loss (OC-CCL), as it creates a cycle by linking the closing frame of the pseudo-video to the opening. See Fig. 6 for an illustration.

3.3. Implementation Detail

Pseudo-video creation. Previous subsections assume that the pseudo-video concatenates the specimen images in an arbitrary order. Intuitively, a smooth transition order should improve SST's performance; in contrast, a non-smooth transition order may degrade SST. So far, using random orders, we have not experienced a degradation. (See Appendix S2.1 for details.) Yet, we believe a dedicated approach to finding a smooth order would make SST more stable, and we leave it as a future work.

In this paper, unless stated otherwise, we implement SST by creating multiple short, two-frame videos. Concretely, given the labeled image-masks pair $(\boldsymbol{x}_0, \boldsymbol{y}_0)$ and N unlabeled images $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$, we create $\{\boldsymbol{x}_0, \boldsymbol{x}_1\}, \cdots, \{\boldsymbol{x}_0, \boldsymbol{x}_N\}$ and apply SST independently to each of them. We have two reasons.

- It eliminates the randomness in creating pseudo-videos while knowing that it would disregard the potential benefit of long-term memory.
- It makes the comparison fair. Conventional models segment each test sample independently, reflecting the real-world online use case where one processes a newly captured image right away. The recent few-shot approaches also process each test image independently. Noticing that considering all test images at once would term the conventional *inductive* setting into a *transductive* one, we decide to process each test image independently.

Fine-tuning and memory bank. Instead of performing full fine-tuning, we apply LORA [27] to fine-tune the decoder and the memory encoder of SAM 2 in a parameter-efficient way. It reduces potential overfitting and speeds up training.

Since we now apply SST to short videos, *i.e.*, $\{\boldsymbol{x}_0, \boldsymbol{x}_n\}$, the memory bank could simply carry the prompt $\boldsymbol{y}_0$ inputted at timestamp 0, making the fine-tuning ineffective. We thus propose the following strategy.

1. We create a palindrome-style cycle $\{\boldsymbol{x}_0, \boldsymbol{x}_n, \boldsymbol{x}_n^\dagger, \boldsymbol{x}_0^\dagger\}$. We use $\dagger$ to denote the duplicated images.
2. In the forward pass, we reset the memory of SAM 2 after the first $\boldsymbol{x}_n$, preventing it from carrying $\boldsymbol{y}_0$ to $\boldsymbol{x}_0^\dagger$.
3. To propagate the mask information from $\boldsymbol{x}_n$ to $\boldsymbol{x}_n^\dagger$, we input the predicted mask $\hat{\boldsymbol{y}}_n$ of the former as the prompt to the latter. This prompt is differentiable.
4. In fine-tuning, SAM 2 needs to learn how to generate a good $\hat{\boldsymbol{y}}_n$ such that $\hat{\boldsymbol{y}}_0^\dagger$ could match the ground-truth $\boldsymbol{y}_0$.

Unless stated otherwise, we fine-tune SAM 2 for each test sample independently, all from the pre-trained weights.

Images with multiple specimens. We assume that each specimen image contains one specimen instance. In practice, if one encounters images with multiple instances, one may first apply object detectors like Grounding DINO [39] to crop out each instance before applying SST.

3.4. Extension to Trait-Based Retrieval

Beyond trait segmentation within the same species, SST can also be used to retrieve specimens having a similar trait (*e.g.*, the white band on the forewing or the orange tiger tails on the hindwing) from other species. Given a query image $\boldsymbol{x}_0$ and a *target* trait $\boldsymbol{y}_0\star$—a single channel in the original $\boldsymbol{y}_0 \in \{0,1\}^{W\times H\times C}$—SST scores each image $\boldsymbol{x}_n$ in the retrieval pool by

1. creating a palindrome-style cycle $\{\boldsymbol{x}_0, \boldsymbol{x}_n, \boldsymbol{x}_n^\dagger, \boldsymbol{x}_0^\dagger\}$;
2. using $\boldsymbol{y}_0\star$ as the prompt and taking the forward pass introduced in Sec. 3.3 to predict $\hat{\boldsymbol{y}}_0^\dagger\star$;
3. calculating the reconstruction quality $\text{IoU}(\boldsymbol{y}_0\star, \hat{\boldsymbol{y}}_0^\dagger\star)$.

Namely, if $\boldsymbol{x}_n$ has the target trait, the mask $\boldsymbol{y}_0\star$ should accurately propagate to $\boldsymbol{x}_n$ and then propagate back to $\boldsymbol{x}_0$.

4. Experiment

4.1. Experimental Setup

Data. We evaluate SST on three specimen data sources.

- **Butterfly:** For fine-grained trait segmentation, we consider the Cambridge Heliconius Collection [36] gathered from various sources[1]. This collection contains 155 butterfly sub-species of the Heliconius genus; each has $4 \sim 14$ distinctive traits to tell itself apart from others. Examples include the tiger tails on the hindwings and white bands on the forewings; some have quite complex, disconnected shapes. We consulted with biologists and the field guide [1] to annotate them. Across specimens of the same sub-species, the mask IDs are consistent. An algorithm needs to not only segment them but also label each with an ID. As this dataset is long-tailed (see Fig. 2), we split the 151 sub-species into two parts: major and minor.

[1] Sources: [30–32, 47, 48, 51, 53–67, 69, 72–75, 88–90]

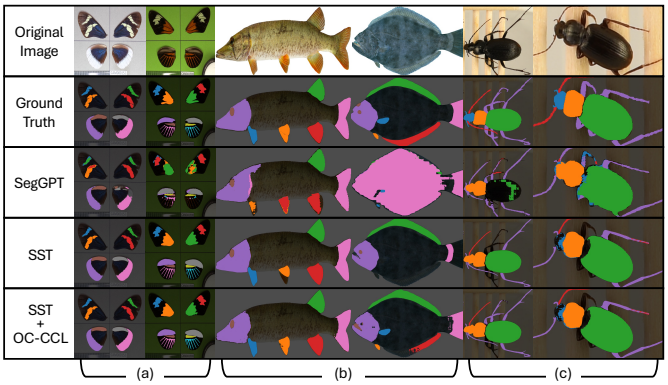

Figure 7. **Qualitative Results:** trait and part segmentation by SST vs. SegGPT [87] on (a) Butterfly [36], (b) Fish [50], and (c) Beetles [22] data.

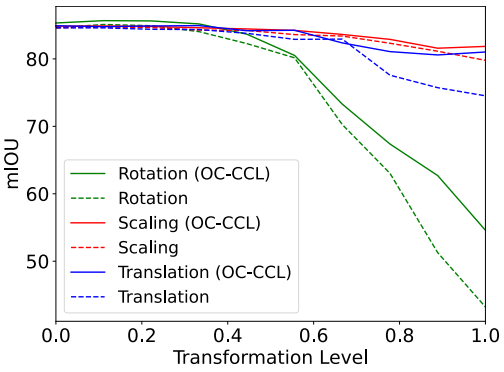

Figure 8. **Fine-Tuning Results.** OC-CCL fine-tuning improves SST's robustness to variations caused by applying different levels of transformations to images.

For each of the five major sub-species with over 250 samples, we randomly sampled 100 specimens as the test set and hand-labeled masks. The remaining samples ($2,831$ in total) are used for training, and we annotated them via a semi-automatic approach: we used our SST to propagate masks, followed by human inspection. Samples with unsatisfactory masks were then hand-labeled. For the minor part, we hand-labeled $2 \sim 3$ samples for each of the 146 sub-species having more than 2 samples.

- **Beetle**: We use the individual image subset of the 2018 NEON-beetles dataset [22]. We hand-labeled 180 specimen images (120/60 for training/testing) with 5 body parts. The antennae and leg parts are quite challenging, with complex, sharp, and thin shapes. Moreover, some specimens have up to 90-degree of body rotations and some have missing or overlapped parts, further increasing the difficulties.
- **Fish**: We use the Fish-Vista dataset [50][2], which contains specimens from over $1,900$ fish species. A subset of specimens ($1,707$/600 for training/testing) were labeled with 9 expert-selected body parts; the labels are consistent across species. We use this to test algorithm robustness: an algorithm needs to segment these parts across species. Some are visually quite distinct, with unique phenotypes.

Evaluation metric. We use the mean IoU (mIoU) as the main metric, averaged over part or trait IDs. In evaluating FSS algorithms (including SST) in a one-shot setting, we sample 10 training specimens with canonical shapes and visually clear traits and report an averaged mIoU over 10 runs of experiments, unless stated otherwise.

Baseline. We compare SST to four representative few-shot segmentation (FSS) methods, PFENet [81], VAT [24], HDMNet [91], and SegGPT [87]. Whenever an algorithm cannot segment multiple classes (*i.e.*, different traits or parts) at once, we segment each class separately. To handle overlapping pixels across classes, we optionally assign each pixel to the highest score label if it improves the mIoU. We also apply two representative many-shot instance segmentation algorithms, YOLOv8 [33] and Mask2Former [15], whenever we have sufficient training samples. Please refer to Appendix S1 for more experimental setup details.

[2]This dataset is comprised of specimen images from various collections: [2–10, 17, 20, 45, 46]

4.2. Main Result

Trait segmentation on Butterfly. As each butterfly subspecies has a distinct set of traits, we evaluate them separately. On each of the five major sub-species, we train a Mask2Former [15] and a YOLOv8 [33] using all training samples. For FSS algorithms, we consider a one-shot setting. On the remaining minor species, we only consider FSS algorithms, again in a one-shot setting. For each minor subspecies, we iterate over the $2 \sim 3$ labeled images, using one for training and the others for testing, and report the average mIoU. Tab. 1 summarizes the results. SST outperforms existing FSS algorithms in a one-shot setting, with a margin of at least 16 mIoU on both major and minor sub-species. Surprisingly, SST even surpasses many-shot algorithms trained with at least 150 samples per sub-species.

Part segmentation. We compare SST to both FSS algorithms (in a one-shot setting) and many-shot algorithms on Fish [50] and Beetle [22] datasets. This is particularly challenging as the algorithm needs to segment the same body parts across species (see Fig. 7), which exhibit huge phenotypic variations. As shown in Tab. 2, most of the FSS methods fail, yet SST can still maintain a fairly high mIoU. We observe further improvement by employing one-shot fine-tuning using OC-CCL, as can be seen in Tab. 2 and Fig. 7.

4.3. Out-of-Distribution (OOD) Robustness

In the actual application of SST on specimen images, sometimes we might encounter OOD cases, where the specimens

Table 1. **Trait Segmentation Results (mIoU).** SST outperforms the other recent FSS methods as well as standard many-shot segmentation models trained on full data.

# of Data	Model	Sub-species	
		Major	Minor
One-Shot	HDMNet [91]	4.2	4.0
	PFENet [81]	8.0	4.2
	VAT [24]	13.5	15.1
	SegGPT [87]	63.3	41.9
	SST (**ours**)	**80.6**	**71.8**
Full	YOLOv8 [33]	71.1	-
	Mask2Former [15]	79.3	-

Table 2. **Part Segmentation Results (mIoU).** SST and fine-tuned SST (SST + OC-CCL) outperform the other FSS models by a large margin.

# of Data	Model	Fish [50]	Beetle [22]
One-Shot	HDMNet [91]	1.5	6.1
	PFENet [81]	3.1	19.1
	VAT [24]	24.6	26.0
	SegGPT [87]	37.5	45.2
	SST (**ours**)	56.8	63.5
	SST + OC-CCL (**ours**)	**61.5**	**66.1**
Full	YOLOv8 [33]	79.8	75.1
	Mask2Former [15]	85.5	83.2

Table 3. **Instance Segmentation.** Our method SST achieves similar results as SOTA FSS methods on object instance segmentation.

Model	mIoU
HDMNet [91]	64.14
PFENet [81]	72.37
VAT [24]	**83.42**
SegGPT [87]	51.33
SST (**ours**)	71.10
SST + OC-CCL (**ours**)	77.76

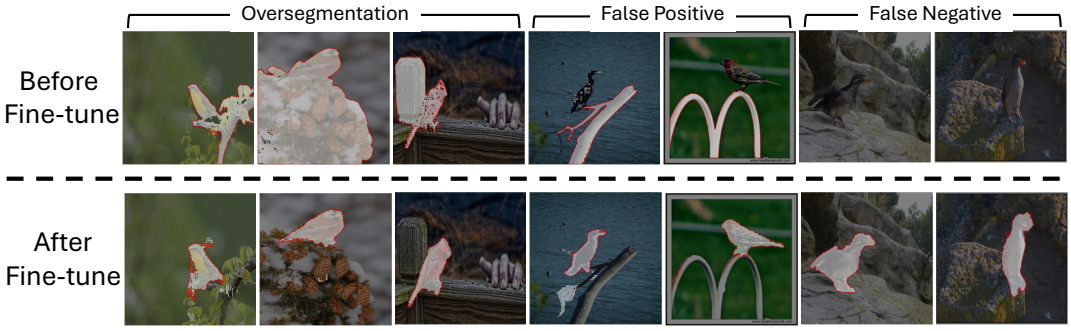

Figure 9. **Before vs. After OC-CCL Fine-Tuning.** Fine-tuning with OC-CCL notably improves SST with merely one labeled training example.

are not captured in standard views. That is, the images might be subjected to rotation, translation, or scaling. Accordingly, we manually apply these transformations to the Butterfly test set to create an OOD robustness task. We define transformation levels from 0.0 to 1.0, corresponding to no transformation and the largest degree of transformation we apply, respectively. At level 1.0, we randomly rotate the image between $-90°$ and $90°$, translate it up to 60% of its height or width, and scale it down to 50% of its original size. We found that SST tends to lose track of the fine-grained details after a certain level of transformations, likely due to the absence of such huge variations (between consecutive frames) in the pre-training data.

To address this, we use OC-CCL to fine-tune the model in a one-shot setting for each test image. OC-CCL consistently improves SST's robustness as seen in Fig. 8. In the extreme rotation cases (the right end of the figure), we boost the mIoU from 40% to over 50%, a more than 10% gain.

4.4. Experiments on Instance Segmentation

Besides fine-grained trait and part segmentation on specimen images, SST can also be applied to object instance segmentation on images taken in the wild. We use the CUB-200-2011 dataset [84] to demonstrate such a capability. Given one random bird image and its segmentation mask, we examine if FSS algorithms can segment all 200 bird species from the remaining images. The results are shown in Tab. 3. As whole object instance segmentation is the original problem domain for most of the com-

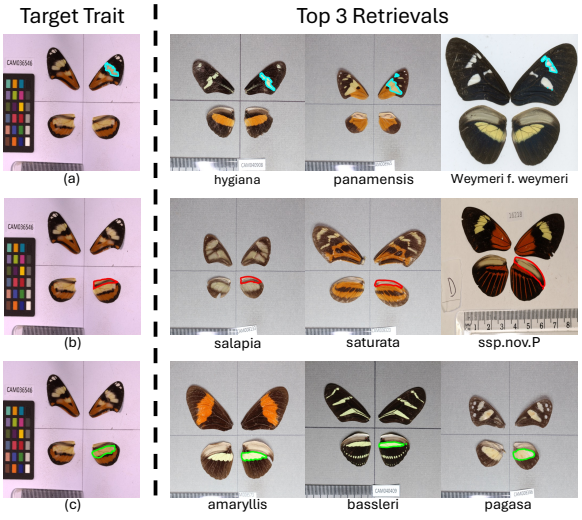

Figure 10. **Trait-Based Retrieval.** Selecting different target traits on the same image retrieves varied lists of sub-species with corresponding traits. Target and retrieved traits are circled in colors.

pared methods, they show much better results than Tab. 1 and Tab. 2. In this domain, SST still achieves a competitive segmentation performance across bird species, even with large variations from image to image. Furthermore, fine-tuning SST with the single training image using OC-CCL again shows significant improvement in segmentation quality. We closely analyze the object instances that originally fail to be correctly segmented by SST and categorize them into 3 failure cases: *Oversegmentation*, where SST correctly segments out the object along with some extra neighboring backgrounds; *False Positive*, where SST falsely segments out an irrelevant object; and *False Negative*, where SST completely fails to segment anything from the picture. As shown in the bottom row of Fig. 9, without using any ground truth masks for the test images, fine-tuning with OC-CCL helps substantially mitigate these issues.

4.5. Experiments on Trait-Based Retrieval

As mentioned in Sec. 3.4, given a target trait, our method can use the reconstruction IoU to find images with similar traits. As shown in Fig. 10, SST + OC-CCL faithfully retrieves sub-species with similar corresponding traits. If we focus on different traits of the same butterfly, we can retrieve different lists of sub-species. For more experiment results and discussions, please see Appendix S4.

5. Conclusion

We introduce **Static Segmentation by Tracking (SST)**, a frustratingly simple approach to fine-grained segmentation on specimen images. By passing non-sequential specimen images into a tracking algorithm like SAM 2, SST demonstrates remarkable trait and part segmentation given merely one labeled image. Our further investigation shows that SST can go beyond specimen images to segment animal instances in the wild. It can also support trait-level retrieval to discover species with similar local parts and patterns.

Acknowledgment

This research is supported in part by grants from the National Science Foundation (OAC-2118240, HDR Institute: Imageomics). The authors are grateful for the generous support of the computational resources from the Ohio Supercomputer Center.

References

[1] La variété des heliconius. https://www.cliniquevetodax.com/Heliconius/index.html. 6

[2] Morphbank: Biological imaging. https://www.morphbank.net/. 7

[3] Multimedia of fish specimen and associated metadata. fishair. https://fishair.org.

[4] Fmnh field museum of natural history (zoology) fish collection. *Field Museum*. https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_fishes.

[5] Great lakes invasives network project. *https://greatlakesinvasives.org/portal/index.php*.

[6] University of wisconsin-madison zoological museum - fish. *http://zoology.wisc.edu/uwzm/*.

[7] Ummz university of michigan museum of zoology, division of fishes. *https://ipt.lsa.umich.edu/resource?r=ummz_fish*.

[8] idigbio. *http://www.idigbio.org/portal*, 2020.

[9] Inhs collections data, 2022.

[10] Jfbm bell atlas. *http://bellatlas.umn.edu/index.php*, 2022. 7

[11] Daniel I Bolnick, Priyanga Amarasekare, Márcio S Araújo, Reinhard Bürger, Jonathan M Levine, Mark Novak, Volker HW Rudolf, Sebastian J Schreiber, Mark C Urban, and David A Vasseur. Why intraspecific trait variation matters in community ecology. *Trends in Ecology & Evolution*, 2011. 1

[12] Ramakant Chandrakar, Rohit Raja, and Rohit Miri. Animal detection based on deep convolutional neural networks with genetic segmentation. *Multimedia Tools and Applications*, 2022. 2, 4

[13] Wei Chen, Yang Liu, Weiping Wang, Erwin Bakker, Theodoros Georgiou, Paul Fieguth, Li Liu, and Michael S Lew. Deep image retrieval: A survey. *arXiv preprint arXiv:2101.11282*, 2021. 3

[14] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Show, match and segment: Joint weakly supervised learning of semantic matching and object co-segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 4

[15] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 3, 4, 7, 8, 13, 14

[16] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 4

[17] Johnson N Daly M. Ohio state university fish division (osum). *Museum of Biological Diversity, The Ohio State University. Occurrence dataset, https://doi.org/10.15468/subsl8*, 2018. 7

[18] Charles Darwin. *On the Origin of Species by Means of Natural Selection*. John Murray, London, 1859. 1

[19] Shiv Ram Dubey. A decade survey of content based image retrieval using deep learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 3

[20] Richard C Edmunds, Baofeng Su, James P Balhoff, B Frank Eames, Wasila M Dahdul, Hilmar Lapp, John G Lundberg, Todd J Vision, Rex A Dunham, Paula M Mabee, et al. Phenoscape: identifying candidate genes for evolutionary phenotypes. *Molecular Biology and Evolution*, 2015. 7

[21] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wies-

beck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 2020. 2, 4

[22] Isadora E. Fluck, Benjamin Baiser, Riley Wolcheski, Isha Chinniah, and Sydne Record. 2018 neon ethanol-preserved ground beetles. https://huggingface.co/datasets/imageomics/2018-NEON-beetles, 2024. 3, 7, 8, 13, 14

[23] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: State of the art. *International Journal of Multimedia Information Retrieval*, 2020. 4

[24] Sunghwan Hong, Seokju Cho, Jisu Nam, and Seungryong Kim. Cost aggregation is all you need for few-shot segmentation. *arXiv preprint arXiv:2112.11685*, 2021. 4, 7, 8, 14

[25] Sunghwan Hong, Seokju Cho, Jisu Nam, and Seungryong Kim. Cost aggregation is all you need for few-shot segmentation. *arXiv preprint arXiv:2112.11685*, 2021. 2, 3

[26] Grant Van Horn and macaodha. iNat challenge 2021 - FGVC8. https://kaggle.com/competitions/inaturalist-2021, 2021. 3

[27] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 6

[28] Yuhao Huang, Xin Yang, Lian Liu, Han Zhou, Ao Chang, Xinrui Zhou, Rusi Chen, Junxuan Yu, Jiongquan Chen, Chaoyu Chen, et al. Segment anything model for medical images? *Medical Image Analysis*, 2024. 4

[29] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 4

[30] Chris Jiggins and Ian Warren. Cambridge butterfly wing collection - Chris Jiggins 2001/2 broods batch 1, 2019. 6

[31] Chris Jiggins and Ian Warren. Cambridge butterfly wing collection - Chris Jiggins 2001/2 broods batch 2, 2019.

[32] Chris Jiggins, Gabriela Montejo-Kovacevich, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 3, 2019. 6

[33] Glenn Jocher, Qiu Jing, and Ayush Chaurasia. Ultralytics yolo. https://github.com/ultralytics/ultralytics. 3, 4, 7, 8, 13

[34] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 4

[35] 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 International Conference on Computer Vision*, 2023. 4

[36] Christopher Lawrence, Elizabeth G. Campolongo, and Neil Rosser. Heliconius collection (cambridge butterfly), 2024. 2, 3, 6, 7, 13, 14, 15, 16

[37] Xiangtai Li, Henghui Ding, Haobo Yuan, Wenwei Zhang, Jiangmiao Pang, Guangliang Cheng, Kai Chen, Ziwei Liu, and Chen Change Loy. Transformer-based visual segmentation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2, 4

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *In Proceedings of the European Conference of Computer Vision*, 2014. 4

[39] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 6

[40] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crnet: Cross-reference networks for few-shot segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 4

[41] Xiangbin Liu, Liping Song, Shuai Liu, and Yudong Zhang. A review of deep-learning-based medical image segmentation methods. *Sustainability*, 2021. 2, 4

[42] Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? In *Advances in Neural Information Processing Systems*, 2021. 3

[43] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, 2021. 2

[44] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 2024. 4

[45] Paula Mabee, James P Balhoff, Wasila M Dahdul, Hilmar Lapp, Peter E Midford, Todd J Vision, and Monte Westerfield. 500,000 fish phenotypes: The new informatics landscape for evolutionary and developmental biology of the vertebrate skeleton. *Journal of Applied Ichthyology*, 2012. 7

[46] Paula M Mabee, Wasila M Dahdul, James P Balhoff, Hilmar Lapp, Prashanti Manda, Josef Uyeda, Todd Vision, and Monte Westerfield. Phenoscape: semantic analysis of organismal traits and genes yields insights in evolutionary biology. In *Application of Semantic Technology in Biodiversity Science*. IOS Press, 2018. 7

[47] Anniina Mattila, Chris Jiggins, and Ian Warren. University of Helsinki butterfly wing collection - Anniina Mattila field caught specimens, 2019. 6

[48] Anniina Mattila, Chris Jiggins, and Ian Warren. University of Helsinki butterfly collection - Anniina Mattila bred specimens, 2019. 6

[49] Maciej A Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: an experimental study. *Medical Image Analysis*, 2023. 4

[50] Kazi Sajeed Mehrab, M. Maruf, Arka Daw, Harish Babu Manogaran, Abhilash Neog, Mridul Khurana, Bahadir Altintas, Yasin Bakış, Elizabeth G Campolongo, Matthew J

Thompson, Xiaojun Wang, Hilmar Lapp, Wei-Lun Chao, Paula M. Mabee, Henry L. Bart Jr., Wasila Dahdul, and Anuj Karpatne. Fish-vista: A multi-purpose dataset for understanding & identification of traits from images, 2024. 3, 7, 8, 13, 15

[51] Joana I. Meier, Patricio Salazar, Gabriela Montejo-Kovacevich, Ian Warren, and Chris Jggins. Cambridge butterfly wing collection - Patricio Salazar PhD wild specimens batch 3, 2020. 6

[52] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2, 4

[53] Gabriela Montejo-Kovacevich, Letitia Cookson, Eva van der Heijden, Ian Warren, David P. Edwards, and Chris Jiggins. Cambridge butterfly collection - loreto, peru 2018, 2019. 6

[54] Gabriela Montejo-Kovacevich, Chris Jiggins, and Ian Warren. Cambridge butterfly wing collection batch 2, 2019.

[55] Gabriela Montejo-Kovacevich, Chris Jiggins, and Ian Warren. Cambridge butterfly wing collection batch 4, 2019.

[56] Gabriela Montejo-Kovacevich, Chris Jiggins, and Ian Warren. Cambridge butterfly wing collection batch 1- version 2, 2019.

[57] Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, Camilo Salazar, Marianne Elias, Imogen Gavins, Eva Wiltshire, Stephen Montgomery, and Owen McMillan. Cambridge and collaborators butterfly wing collection batch 10, 2019.

[58] Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 5, 2019.

[59] Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 6, 2019.

[60] Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 7, 2019.

[61] Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 8, 2019.

[62] Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, Eva Wiltshire, and Imogen Gavins. Cambridge butterfly wing collection batch 9, 2019.

[63] Gabriela Montejo-Kovacevich, Letitia Cookson, Eva van der Heijden, Ian Warren, David P. Edwards, and Chris Jiggins. Cambridge butterfly collection - Loreto, Peru 2018 batch2, 2020.

[64] Gabriela Montejo-Kovacevich, Letitia Cookson, Eva van der Heijden, Ian Warren, David P. Edwards, and Chris Jiggins. Cambridge butterfly collection - Loreto, Peru 2018 batch3, 2020.

[65] Gabriela Montejo-Kovacevich, Eva van der Heijden, and Chris Jiggins. Cambridge butterfly collection - GMK Broods Ikiam 2018, 2020.

[66] Gabriela Montejo-Kovacevich, Eva van der Heijden, Nicola Nadeau, and Chris Jiggins. Cambridge butterfly wing collection batch 10, 2020.

[67] Gabriela Montejo-Kovacevich, Quentin Paynter, and Amin Ghane. Heliconius erato cyrbia, Cook Islands (New Zealand) 2016, 2019, 2021, 2021. 6

[68] Tai-Yu Pan, Qing Liu, Wei-Lun Chao, and Brian Price. Towards open-world segmentation of parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 4

[69] Erika Pinheiro de Castro, Christopher Jiggins, Karina Lucas da Silva-Brand0̆0e3o, Andre Victor Lucci Freitas, Marcio Zikan Cardoso, Eva Van Der Heijden, Joana Meier, and Ian Warren. Brazilian Butterflies Collected December 2020 to January 2021, 2022. 6

[70] 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. 2, 3, 4, 5

[71] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 4

[72] Camilo Salazar, Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Imogen Gavins. Camilo Salazar and Cambridge butterfly wing collection batch 1, 2019. 6

[73] Patricio Salazar, Gabriela Montejo-Kovacevich, Ian Warren, and Chris Jiggins. Cambridge butterfly wing collection - Patricio Salazar PhD wild and bred specimens batch 1, 2018.

[74] Patricio Salazar, Gabriela Montejo-Kovacevich, Ian Warren, and Chris Jiggins. Cambridge butterfly wing collection - Patricio Salazar PhD wild and bred specimens batch 2, 2019.

[75] Patricio A. Salazar, Nicola Nadeau, Gabriela Montejo-Kovacevich, and Chris Jiggins. Sheffield butterfly wing collection - Patricio Salazar, Nicola Nadeau, Ikiam broods batch 1 and 2, 2020. 6

[76] Xinru Shan and Chaoning Zhang. Robustness of segment anything model (sam) for autonomous driving in adverse weather conditions. *arXiv preprint arXiv:2306.13290*, 2023. 4

[77] Pamela S Soltis. Digitization of herbaria enables novel research. *American journal of botany*, 104(9):1281–1284, 2017. 4

[78] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, 2020. 3

[79] Lv Tang, Haoke Xiao, and Bo Li. Can sam segment anything? when sam meets camouflaged object detection. *arXiv preprint arXiv:2304.04709*, 2023. 4

[80] Shehbaz Tariq, Brian Estadimas Arfeto, Chaoning Zhang, and Hyundong Shin. Segment anything meets semantic communication. *arXiv preprint arXiv:2306.02094*, 2023. 4

[81] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2, 3, 4, 7, 8, 14

[82] Cyrille Violle, Brian J. Enquist, Brian J. McGill, Lin Jiang, Céline H. Albert, Catherine Hulshof, Vincent Jung, and Julie Messier. The return of the variance: intraspecific variability in community ecology. *Trends in Ecology & Evolution*, 2012. 1

[83] Huy V Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *In Proceedings of the European Conference of Computer Vision*. Springer, 2020. 4

[84] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 3, 8, 14

[85] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 3

[86] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[87] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Towards segmenting everything in context. In *Proceedings of the IEEE International Conference on Computer Vision*, 2023. 2, 3, 7, 8, 14

[88] Ian Warren and Chris Jiggins. Miscellaneous Heliconius wing photographs (2001-2019) Part 1, 2019. 6

[89] Ian Warren and Chris Jiggins. Miscellaneous Heliconius wing photographs (2001-2019) Part 2, 2019.

[90] Ian Warren and Chris Jiggins. Miscellaneous Heliconius wing photographs (2001-2019) Part 3, 2019. 6

[91] Weiyi Xue, Fan Lu, and Guang Chen. Hdmnet: A hierarchical matching network with double attention for large-scale outdoor lidar point cloud registration. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2024. 4, 7, 8, 14

[92] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys*, 2006. 2

[93] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 4

Static Segmentation by Tracking: A Frustratingly Label-Efficient Approach to Fine-Grained Segmentation

Supplementary Material

Table S1. **Dataset Statistics** for Butterfly [36], Fish [50], and Beetle [22] dataset.

	Butterfly		Fish	Beetle
	Major	**Minor**		
# of Classes	5	146	465	12
Total Train	2831	-	1707	120
Total Test	500	313	600	60

S1. Dataset Statistics and Evaluation Details

We put the general statistics for Butterfly [36], Fish [50], and Beetle [22] datasets in Tab. S1.

Butterfly. The Cambridge Butterfly [36] dataset includes 151 sub-species in total. We split it into two parts, major and minor, based on the available data sample for each sub-species. The major part has 5 different sub-species, with 2,831 semi-automatically labeled training samples and 500 hand-labeled test samples in total. We take all 2,831 training data samples to train standard segmentation models, Mask2Former [15] and YOLOv8 [33], for each sub-species. To evaluate few-shot segmentation models, we sample one random specimen from the train set for each sub-species, and evaluate the performance on all test data of the same sub-species. The minor part has 146 sub-species with 313 hand-labeled test images in total, which is intended for the one-shot segmentation task. For this part, we only test on few-shot segmentation models in the same fashion as we evaluate the major sub-species.

Fish. The Fish-Vista [50] dataset has 465 different species of fish, containing 1,707 training samples and 600 testing samples in total. As all species share a common set of 9 segmentation classes (*e.g.* head, eye, tail, adipose fin, caudal fin, etc.), we are able to train standard segmentation models on all 1,707 samples from the train set across all species. For few-shot segmentation models, we select 10 representative examples from the training set as reference, and use each of them to evaluate the one-shot performance.

Beetle. The Beetle [22] dataset consists of beetles of 12 different species. For each species, we hand-label 15 images in total, taking 10 as the train set and 5 as the test set. Each beetle species shares 5 common segmentation classes: head, pronotum, elytra, antenna, and legs. For the standard segmentation model, we train across all training data across species. For few-shot segmentation methods, we randomly sample one example from the 120 training samples and test the segmentation quality on all 60 test data across species.

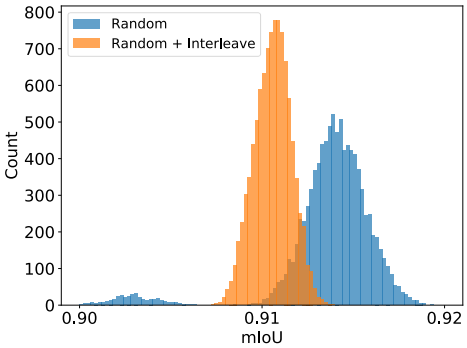

Figure S1. **SST Performance on Different Inference Variants.** Optimizing the frame ordering improves the stability when using long video sequences for inference.

S2. Additional Analysis on SST

S2.1. Analysis on Inference Variants

In the main paper, we evaluate our method using a single test image at a time to compare it against other few-shot segmentation models, ensuring a fair comparison. However, as mentioned in Section 3.3, it is also possible to concatenate all test images and process them together using SST. We observe that using random orders does not significantly degrade performance, though a more carefully designed algorithm could make SST more stable.

To demonstrate this, we conduct a toy experiment on the lativitta sub-species from the Butterfly [36] dataset. We first randomly sample 10 butterflies from the test set and generate 10,000 unique orderings for the 10 images. We then put all ten images in a sequence based on each ordering and evaluate SST after propagating through the entire sequence. We average the mIoU performance across each ordering and plot it against a histogram as shown in the blue part of Fig. S1. There appear to be two peaks in the distribution, but the overall influence is limited (mIoU from 90% to 92%). We then experiment with a slightly improved ordering strategy by interleaving each test image with the reference image, so that the reference information can be retained even if some frames lose the track. As shown in the orange part of Fig. S1, although random ordering only has a limited influence on the performance, interleaving the test images further reduces the standard deviation in mIoU. Thus we conclude that finding the optimal sequence can indeed help with the stability of video inference, and we plan to explore the ordering design as a future work.

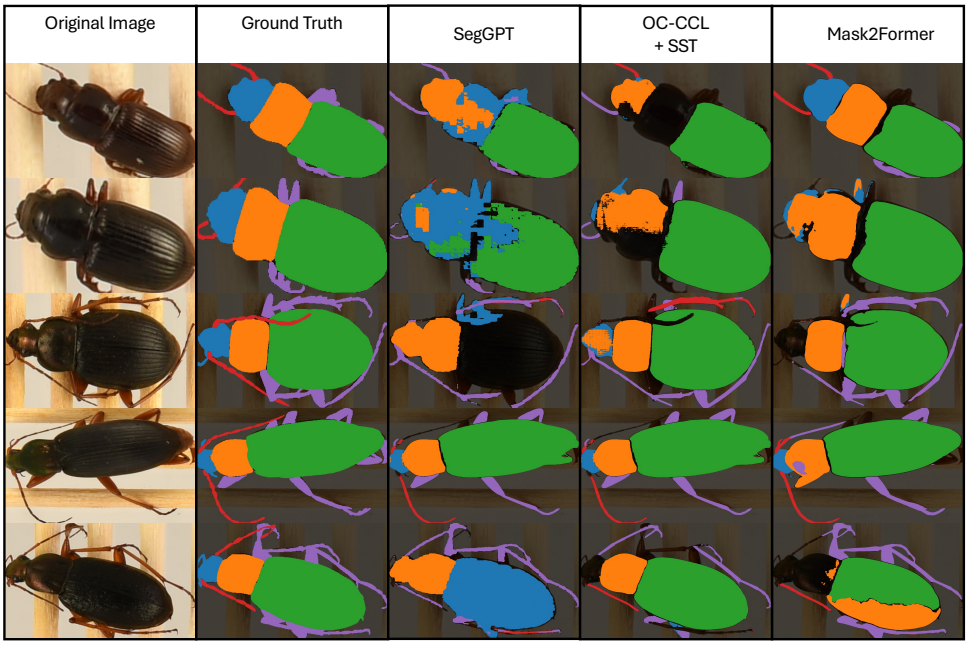

Figure S2. **Qualitative Results on Beetle [22].**

Table S2. **Multi-Shot on Instance Segmentation.** SST can also benefit from multiple examples.

Model	1-shot	5-shot
HDMNet [91]	64.1	66.3
PFENet [81]	72.4	72.8
VAT [24]	**83.4**	**85.3**
SegGPT [87]	51.3	78.8
SST (**ours**)	71.1	77.9

S2.2. Multi-Shot Setting

Similar to the other few-shot segmentation algorithms, SST can also benefit under the multi-shot setting. For 5-shot evaluation, we adopt a naive design, where five labeled image mask pairs are put in the beginning, and a sequence is created independently for each of the incoming images. In other words, given labeled reference pairs (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) and unlabeled target images $\{x_5, \ldots, x_N\}$, we create sequence $\{x_0, \ldots, x_4, x_5\}, \ldots, \{x_0, \ldots, x_4, x_N\}$ and apply SST to each of them. As shown in Tab. S2, SST's 5-shot performance on CUB-200-2011 [84] bird instance segmentation task scores 77.9 average mIoU, with a 6.8 improvement. It can also be noticed that SegGPT benefits significantly from more training samples. We also aim to design a better multi-shot mechanism for SST as a future work.

S3. Qualitative Results on Fine-Tuned Models

To compare the performance of different methods, we show more qualitative results on a more diverse set of butterflies, fish, and beetle species, see Fig. S2, Fig. S3, and Fig. S4. For each column, we keep the same setting as in the main paper. For both SegGPT [87] and SST + OC-CCL, we select one random image from the training set as a reference and evaluate the segmentation quality on the target image. The quality is demonstrated in the third and fourth columns of the figures. We also show the segmentation quality of Mask2Former [15] in the last column of Fig. S2 and Fig. S4, which is trained on the entire available training dataset. We omit the Mask2Former column for the Butterfly [36] dataset as there aren't enough data samples to train a full standard segmentation model for most of these sub-species.

S4. Additional Trait-Based Retrieval Results

We originally demonstrate SST's ability to do trait-base retrieval in Section 3.4 and 4.5 in the main paper. Here, we show more trait-based retrieval results using SST with a diverse range of sub-species. Given a target trait on any sub-species, as outlined in red in the left-most column of Fig. S5, SST can reliably retrieve sub-species that share similar traits, as outlined in cyan in Fig. S5.

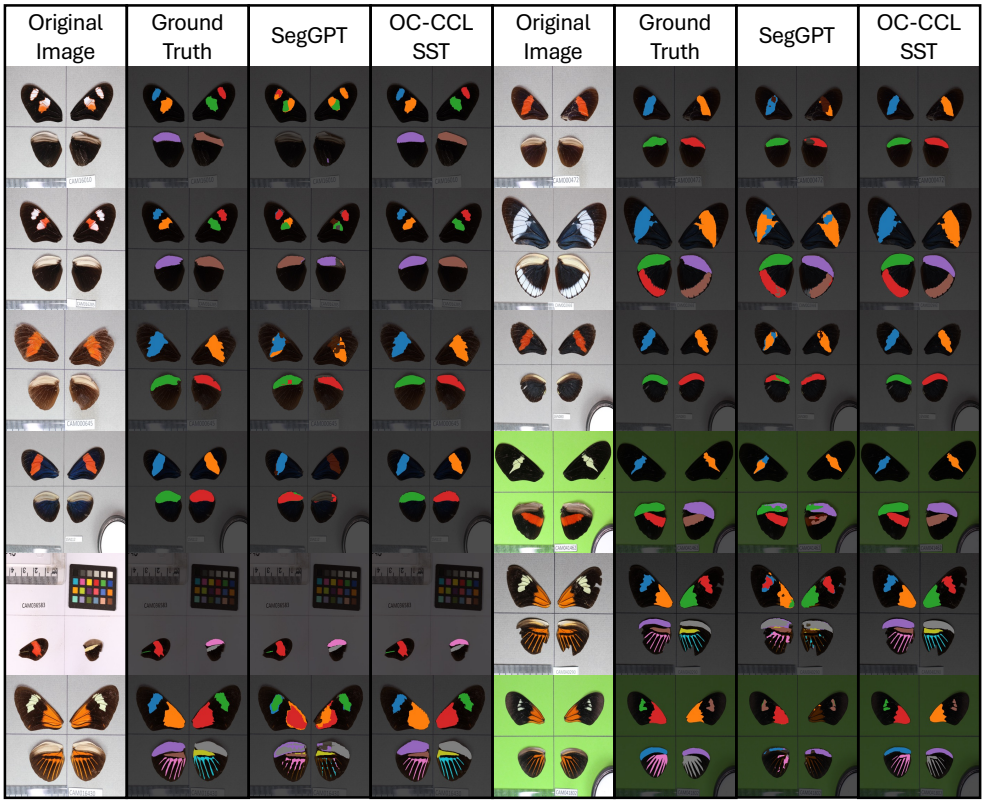

Figure S3. **Qualitative Results on Butterfly [36].**

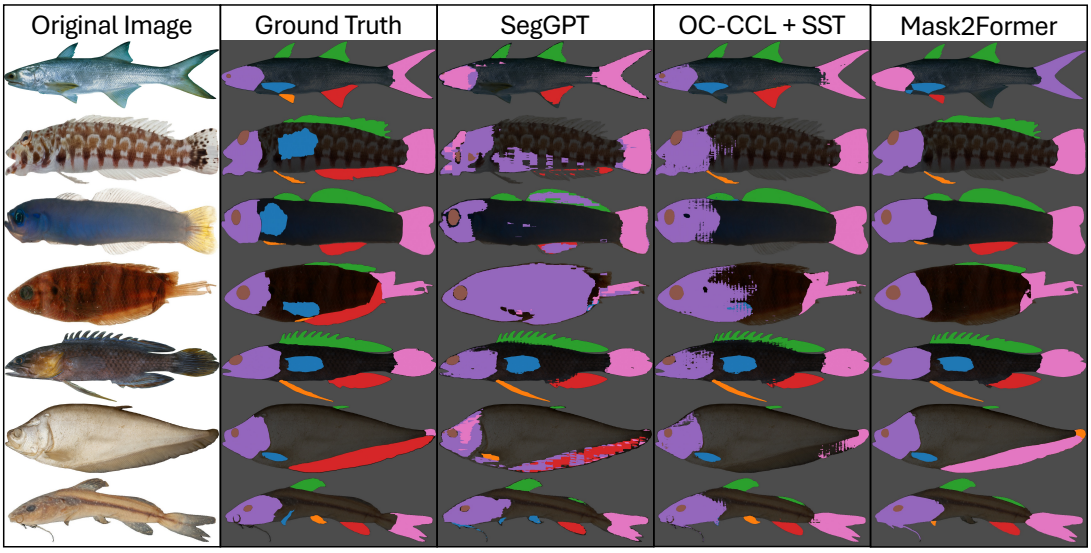

Figure S4. **Qualitative Results on Fish [50].**

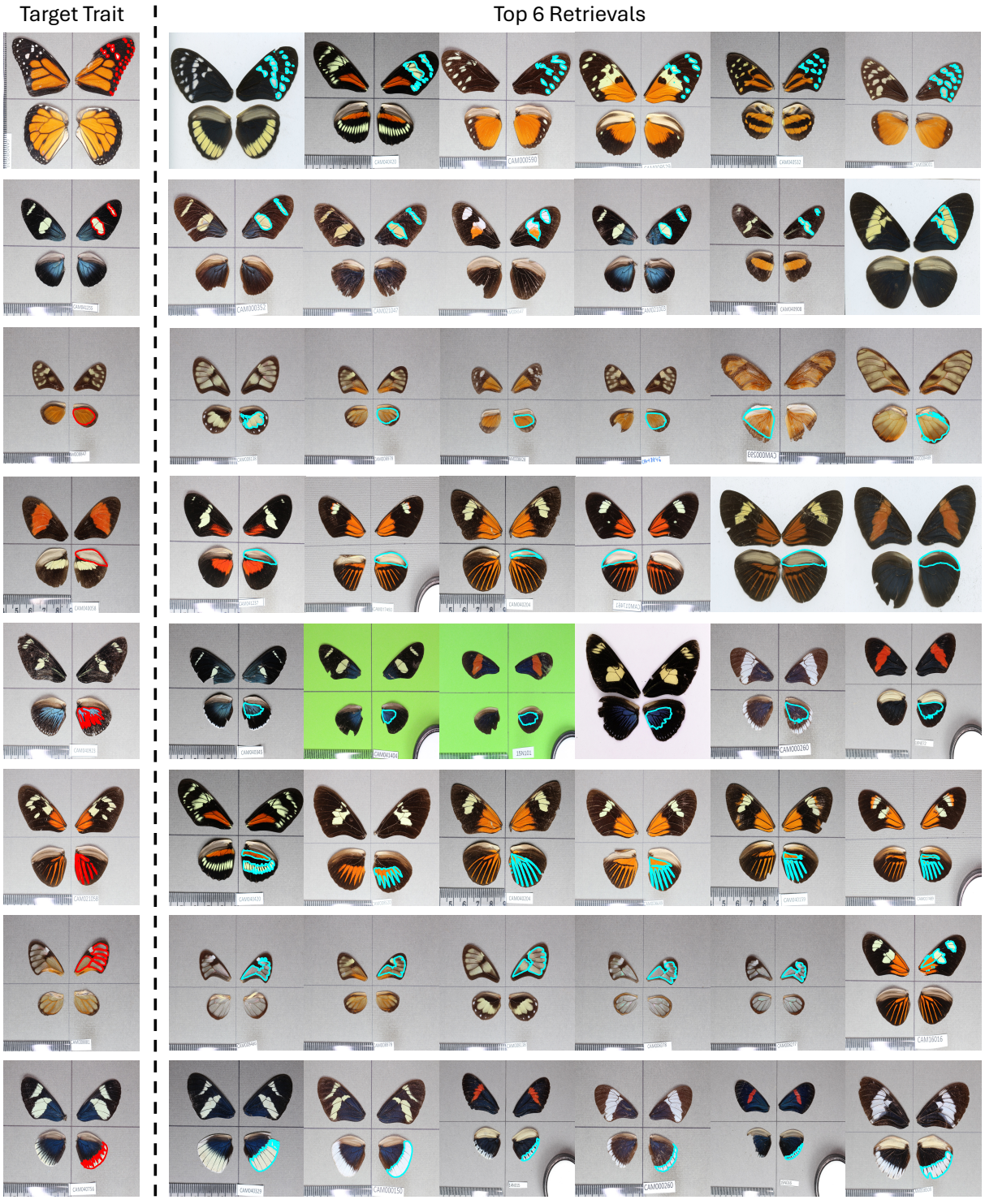

Figure S5. **Qualitative Results for Trait Retrieval on Butterfly [36].** Target trait is outlined in **red**, the retrieved traits are outlined in **cyan**.