DINOSim: Zero-Shot Object Detection and Semantic Segmentation on Electron Microscopy Images

Aitor González-Marfil[1,2], Estibaliz Gómez-de-Mariscal[3,4,5], and Ignacio Arganda-Carreras[1,2,6,7]

[1]Dept. of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), San Sebastian, Spain
[2]Donostia International Physics Center (DIPC), San Sebastian, Spain
[3]Instituto Gulbenkian de Ciência, Oeiras, Portugal
[4]Gulbenkian Institute for Molecular Medicine, Oeiras, Portugal
[5]António Xavier Institute of Chemical and Biological Technology, Universidade Nova de Lisboa
[6]Biofisika Institute (CSIC-UPV/EHU), Leioa, Spain
[7]IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

We present DINOSim, a novel approach leveraging the DINOv2 pretrained encoder for zero-shot object detection and segmentation in electron microscopy datasets. By exploiting semantic embeddings, DINOSim generates pseudo-labels from patch distances to a user-selected reference, which are subsequently employed in a k-nearest neighbors framework for inference. Our method effectively detects and segments previously unseen objects in electron microscopy images without additional fine-tuning or prompt engineering. We also investigate the impact of prompt selection and model size on accuracy and generalization. To promote accessibility, we developed an open-source Napari plugin, enabling streamlined application in scientific research. DINOSim offers a flexible and efficient solution for object detection in resource-constrained settings, addressing a critical gap in bioimage analysis.

Electron Microscopy | Object Detection | Image Segmentation | Foundation Models

Correspondence: ***aitor.gonzalezm@ehu.eus***

Introduction

Obtaining labeled data is a costly and time-intensive process in many computer vision applications, posing a significant challenge for training fully supervised models. Few-shot learning (1) has emerged as a solution to this problem, enabling models to effectively learn target tasks using a limited amount of labeled data. Within the scope of few-shot learning, zero-shot learning eliminates the need for labeled examples altogether (2), making it especially valuable in fields where extensive labeled datasets are scarce.

A key enabler of these advancements is the emergence of foundation models (3), which are large-scale deep learning models pre-trained on vast datasets and later adapted for specific tasks. Transformers (4), particularly the Vision Transformer (ViT) (5), have become a preferred architecture for foundation models in both natural language processing (NLP) and computer vision. These models leverage self-supervised learning (SSL) (6, 7), a paradigm that effectively utilizes large amounts of unlabeled data to train models capable of performing various tasks, such as classification and segmentation in computer vision. As a result, foundation models exhibit strong zero-shot and few-shot capabilities (8), demonstrating remarkable generalization to tasks and data distributions outside their original training domains.

A prominent example of a foundation model in computer vision is the Segment Anything Model (SAM) (9). SAM combines an image encoder with a prompting system that uses points, bounding boxes or masks to support object segmentation in images or videos (SAM2 (10)). However, SAM's prompting system lacks the flexibility of natural language prompts, which could offer more detailed and descriptive segmentation instructions. By default, SAM performs instance segmentation, requiring separate prompts for each object of interest. This limitation becomes cumbersome when dealing with multiple objects in an image or across a set of images. While some efforts have been made to integrate text-based prompts with SAM (11), these approaches often involve prompt engineering, which can introduce additional complexity. Other strategies combine SAM-like models with object detection modules (12). However, these methods typically require fine-tuning the model for specific tasks, thereby reducing the versatility of SAM's original design and limiting its ability to dynamically select different objects.

Another notable foundation model in computer vision is DINO (self-DIstillation with NO labels) (13), which employs a teacher-student paradigm. In this setup, a "teacher" model guides the learning of a "student" model, enabling iterative learning and progressive improvement. The latest version, DINOv2 (14), introduces a refined training process that produces higher-quality features. These features excel at capturing semantic relationships within images, ensuring that similar content is represented consistently, even across different domains or styles. The DINOv2 image encoder has demonstrated outstanding performance in tasks such as segmentation, classification, and depth estimation, achieving these results with minimal fine-tuning or through the addition of task-specific components like classification heads or segmentation decoders (14, 15).

In life sciences, acquiring and annotating bioimage data is often both expensive and time-consuming (16). Bioimages exhibit significant variability due to differences in the imaging protocols, including staining techniques, imaging de-

vice calibration, and the inherent diversity of biological samples (17). These factors make it challenging –and sometimes impractical– to compile large, comprehensive datasets. For instance, while the CEM1.5M dataset (18) aims provides a large and diverse dataset for mitochondria segmentation in electron microscopy images, many subfields continue to struggle with insufficient data for rare scenarios and pathologies. This scarcity of high-quality training data is a major obstacle in developing foundation models tailored to biological data, as the performance of these models is inherently limited by the quality and diversity of their training datasets (19).

In NLP, the large language models (LLMs) face similar challenges in specialized fields like biomedicine or microscopy, where domain-specific training data is scarce (20, 21). Although fine-tuning with domain-specific data can improve performance (22–24), this approach requires considerable computational and data resources. General-purpose foundation models, such as SAM, may not always outperform specialized models in task-specific performance, but they excel in adaptability and generalization without requiring additional training. This adaptability makes such models particularly valuable in complex fields where acquiring high-quality data is difficult to obtain.

We hypothesize that the embeddings generated by DINOv2 can be effectively utilized for zero-shot object detection and segmentation of previously unseen objects in bioimages using point-based prompts. By leveraging DINOv2's capabilities, our approach targets the detection of objects similar to a user-selected reference, even in image domains that were not part of DINOv2's training dataset. Our method, named DINOSim, eliminates the need for additional training, fine-tuning of the foundation model, or prompt engineering.

The main contributions of this work are as follows:

1. We demonstrate how embeddings from a foundation model (DINOv2) enable zero-shot segmentation and detection in two electron microscopy image datasets, without the need for fine-tuning.

2. We evaluate the trade-offs between model size and detection performance.

3. We analyze the impact of prompt selection and quantity on detection stability and accuracy.

4. We provide an open-source and user-friendly Napari plugin (https://github.com/AAitorG/napari-dinoSim), offering an intuitive and interactive graphical interface in widely used software by the life sciences community.

Related Work

Zero-shot segmentation and object detection. Zero-shot learning is an advanced extension of few-shot learning that eliminates the need for labeled examples, enabling the segmentation or detection of new categories of objects without relying on annotated training data (1, 2). Like other few-shot learning methods, zero-shot learning leverages prior knowledge, often through techniques such as transfer learning. These approaches are typically classified into two main categories: projection-based methods and generative methods.

Projection-based methods. Projection-based methods map visual features into a semantic space, such as word embeddings, to infer relationships between known and unknown classes. A notable example is Grounding DINO (25), which generates bounding boxes based on textual prompts. When combined with SAM, Grounded SAM (11) uses these bounding boxes as prompts to refine segmentations for the specified textual prompt. Another example is CLIPSeg (26), which leverages the contrastive language-image pre-training (CLIP) model (27) along with a custom segmentation decoder to generate segmentations from textual prompts. Additionally, CLIPSeg supports image-based prompts, where a labeled mask could highlight the target object. However, in scenarios where labeled masks are used as prompts, the approach transitions to one-shot learning rather than zero-shot learning.

Generative methods. Generative methods create features for unseen categories by learning from labeled data and synthesizing pseudo-labeled samples for new classes. These approaches often employ text or word embeddings, such as Word2Vec (28). For instance, ZS3Net (29) trains a generator to map these embeddings into features that represent unseen classes, enabling segmentation or detection tasks without explicit labels.

Interactive learning. Interactive learning iteratively refines model outputs via user interactions, while weakly supervised learning relies on training models using incomplete or noisy labels. A cost-effective example of weakly supervised learning involves using small models, such as random forests (30), to classify predefined image features like Sobel or Gaussian filters (31, 32). Convpaint (33) extends this concept by incorporating image encoder features, such as those from VGG (34) or DINOv2, with an interactive learning framework. In this setup, users iteratively improve labels by adding annotations (e.g., doodles) for each class, thereby enhancing segmentation quality in a user-guided manner.

Methods

DINOSim introduces a zero-shot learning approach that leverages the embeddings from DINOv2, which effectively capture the semantic structure of an image. The minimum input required by DINOSim consists of an image and a point prompt that serves as reference for guiding the segmentation (Figure 1a). The image is first mapped into the embedding space using DINOv2, and the coordinates of the point prompt are projected onto that space to obtain its corresponding vector representation. Given this reference vector, DINOSim computes the distance between each position of the embedding and the reference vector. This distance quantifies the semantic similarity between each pixel and the prompt, en-

abling the pseudo-labeling of each vector in the embedding space with its corresponding distance value.
The pseudo-labeled reference embedding is then used to estimate the similarity values between the embedding of new images and the reference vector. This is achieved through a k-nearest neighbor (k-NN) approximation which enhances robustness (Figure 1c). Finally, the distance map is filtered, upsampled to match original image resolution, and thresholded to generate the final segmentation mask (Figure 1b). The overall method is illustrated in Figure 1.

Reference embeddings. Using the DINOv2 image encoder, we extract embeddings from a set of T reference images of size of $H \times W$, each associated with at least one prompt. Since DINOv2 is a ViT, input images are resized to $H_p \times W_p$, ensuring compatibility with the model's patch size. For each reference image $r \in \{1, \ldots, T\}$, its embedding is represented as $E_r \in \mathbb{R}^{\mathcal{H} \times \mathcal{W} \times \mathcal{F}}$, where $\mathcal{H}$ and $\mathcal{W}$ denote the spatial dimensions of the embedding, and $\mathcal{F}$ represents its feature depth.

Reference distance maps. For a given reference image r, a user-provided set of point-like prompts defines the target elements or regions of interest (ROIs) at spatial coordinates $P_r = \{(x_1, y_1), (x_2, y_2), \cdots, (x_{n_r}, y_{n_r})\}$. To estimate the feature vector of the prompt $i \in [1, ..., n_r]$ at location (x_i, y_i), within the embedding E_r, we use bilinear interpolation of the four nearest feature vectors of the point (x_i, y_i) mapped to the embedding space. This way, the corresponding embedding vectors $\{e_1^r, e_2^r, \cdots, e_{n_r}^r\}$ are extracted. The reference vector is computed as the mean of all extracted vectors across all prompts and images:

$$e_{ref} = \frac{1}{\sum_{r=1}^{T} n_r} \sum_{r=1}^{T} \sum_{i=1}^{n_r} e_i^r. \quad \textbf{(1)}$$

The reference distance map $D_r \in \mathbb{R}^{\mathcal{H} \times \mathcal{W}}$ for each image r is computed as the Euclidean distance between every vector in E_r and the reference vector e_{ref} (Figure 1a):

$$D_r[x, y] = ||e_{ref} - E_r[x, y]||_2, \quad [x, y] \in \mathcal{H} \times \mathcal{W}. \quad \textbf{(2)}$$

Here, D_r represents the proximity of each pixel to the reference pixels in the embedding space, providing a pseudo-labeling map for subsequent segmentation.

Noise reduction and normalization. Each distance map D_r is smoothed using a Gaussian filter $G(\cdot; kernel = 3, \sigma = 1)$. To mitigate outliers, we apply quantile normalization with a 1% clipping factor, setting the lower and upper clipping thresholds to the 1st and 99th percentiles across all D_r values. The clipped maps, D'_r, are then normalized to the $[0, 1]$ range using the global minimum and maximum values:

$$D_r^* = \frac{D'_r - min_{\forall r}(D'_r)}{max_{\forall r}(D'_r) - min_{\forall r}(D'_r)} \quad \textbf{(3)}$$

The resulting set of normalized distance maps, $\{D_1^*, D_2^*, \cdots, D_T^*\}$, is stored for use in the next step.

k-NN distance maps. Given a new image and its embedding E_{new}, the embeddings from the reference set $\{E_1, E_2, \cdots, E_T\}$ are collectively used to determine the k-nearest neighbors for each coordinate in E_{new}. The predicted distance value is computed as:

$$D_{new}^*[x, y] = \frac{1}{k} \sum_{i=1}^{k} D_{i^*}^*[x_{i^*}, y_{i^*}] \quad \textbf{(4)}$$

where $E_{i^*}[x_{i^*}, y_{i^*}]$ denotes the i-th nearest neighbor between $E_{new}[x, y]$ among all vectors in $\{E_1, E_2, \cdots, E_T\}$ (Figure 1c).

Post-processing. To refine the predicted distance map D_{new}^*, we first apply a Gaussian filter $G(\cdot; kernel = 3, \sigma = 1)$. The smoothed map is then upsampled to the original image resolution ($upsample(\cdot; H \times W)$) using bilinear interpolation. The final binary segmentation mask M is obtained by thresholding the upsampled distance map with a threshold τ (empirically defined):

$$M[x, y] = \begin{cases} 1 & upsample(G(D_{new}^*))[x, y] \leq \tau \\ 0 & otherwise \end{cases} \quad \textbf{(5)}$$

Tiling strategy. To efficiently process large images of size $H \times W$, we divide them into tiles of size $H_t \times W_t$, with the minimum overlap between tiles based in the relation between the total image size and the tile size. Each tile is processed independently through DINOSim up to, but not including, the post-processing step. Before post-processing, the full-size distance map D_{new}^* is reconstructed by merging the individual tile predictions. In regions where tiles overlap, pixel values are averaged. Finally, post-processing is applied, and the binary segmentation mask (Eq. 5) is computed at the original image resolution $H \times W$.

Benchmarking

Benchmarking against fully supervised methods. To assess the feasibility of using DINOv2 image embeddings for microscopy images, we trained multiple models in a fully supervised manner. Specifically, we implemented: k-nearest neighbors (k-NN) with $k = 5$, Naive Bayes (NB), a decision tree (DT), a random forest (RF) and a multi-layer perceptron (MLP) with a single hidden layer of 200 neurons. These models were trained to classify embedding vectors extracted from DINOv2 ViT-g. Given an embedding vector ($E[x, y]$), the models learned to predict its corresponding class.
To generate training labels, we downsampled ground truth semantic segmentation masks to the embedding resolution ($\mathcal{H} \times \mathcal{W}$) using nearest-neighbor interpolation, ensuring that each embedding vector had an associated label. For final predictions at the original image resolution ($H \times W$), we applyed the same post-processing pipeline as in DINOSim.
For comparison with state-of-the-art fully supervised segmentation methods, we trained a U-Net-like model (35) with four levels, ReLU activations, batch normalization (36), 16 initial filters, and batch size of six. The model was trained

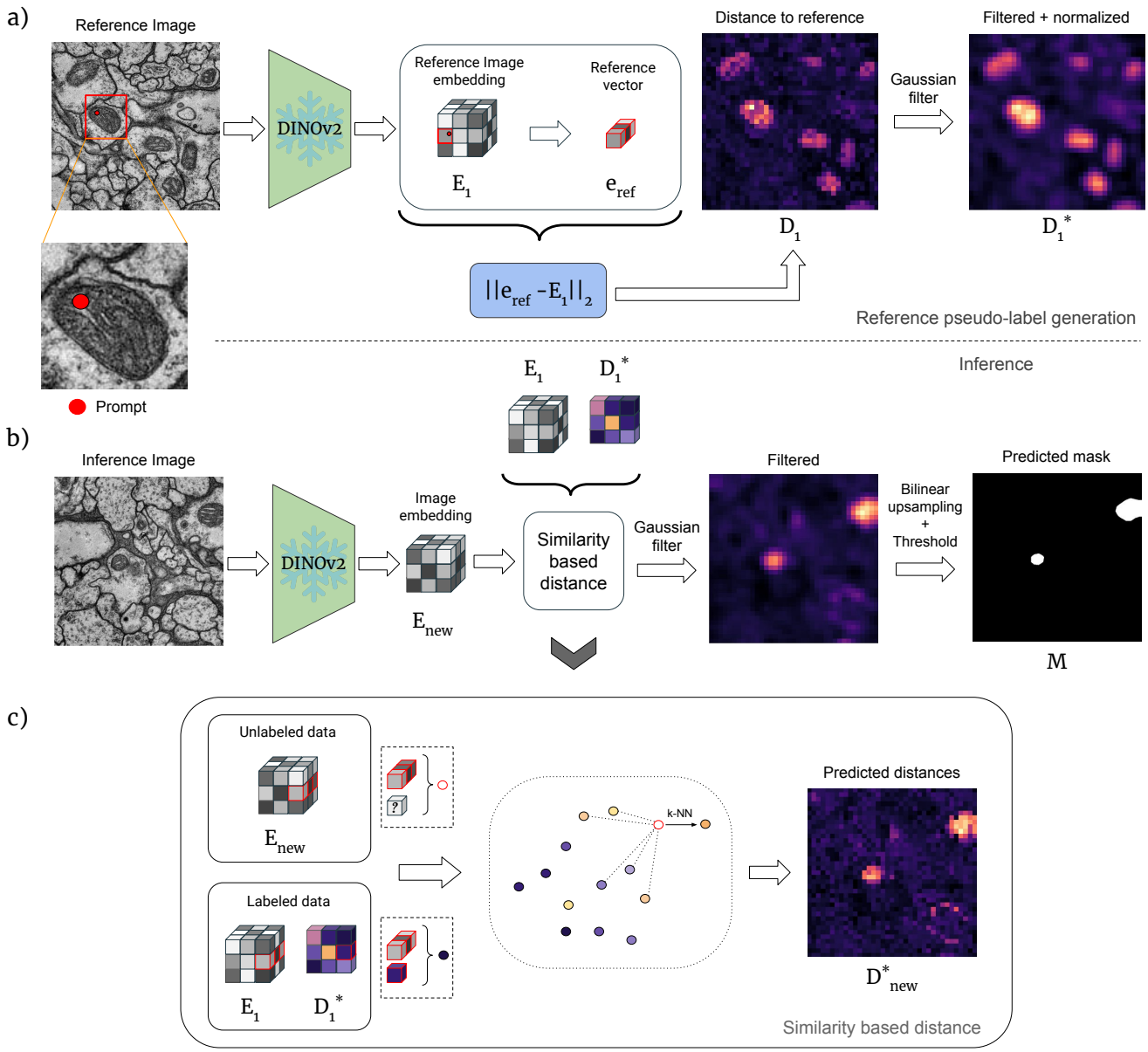

Fig. 1. Overview of the DINOSim zero-shot segmentation method. **(a)** Pseudo-label generation workflow: The image embeddings are extracted using DINOv2. To construct the reference distance map, the Euclidean distance is computed between each patch in the embedding space and the reference vector corresponding to the selected prompt's coordinates. In the resulting distance map, D, lighter areas indicate closer proximity to the reference. The distance map is then smoothed using a Gaussian filter and normalized to obtain D^*. **(b)** Inference workflow: The processed distance map D^* and the embeddings from the training image set ($\{E_1, E_2, \ldots, E_T\}$, where $T = 1$ in this case) are used to compute a distance map for a new image via k-NN estimation. **(c)** Using k-NN, the new distance map is computed as the average of the distance values from the k-nearest embedding vectors for each pixel in the embedding space. The resulting distance map is then refined using a Gaussian filter, yielding D^*_{new}. The final segmentation mask, M, is obtained by upscaling and thresholding the smoothed distance map.

for 200 epochs using the binary cross-entropy loss function and the AdamW (37) optimizer with a weight decay of $1e-5$ and one-cycle learning rate scheduler (38), reaching a maximum learning rate of $1e-3$. Unlike DINOv2, our U-Net was trained on 256×256 pixel image crops.

To ensure statistical robustness, we trained each fully supervised model ten times and report the mean and standard deviation of the accuracy metrics.

Benchmarking against few-shot and zero-shot methods. We compared DINOSim with state-of-the-art few-shot and zero-shot segmentation and detection methods. Specifically, we evaluated CLIPSeg (26) and Convpaint (33) for semantic segmentation, as well as Grounding DINO (25) for object detection. For a fair comparison, we extracted bounding boxes from segmentation masks following the same procedure across all methods. Results are summarized in Tables 2 and 3.

Textual prompts. For text-based methods, we initially tested class-related terms such us "mitochondria" and "organelle" but found that more descriptive prompts, such as "black object" or "black round object", yielded better results. Consequently, these more descriptive prompts were used for Grounding DINO and CLIPSeg respectively.

Image prompts. For CLIPSeg, we also explored image-based prompting. We tested various strategies, including blurring or darkening the background, as suggested in the original paper. The best results were obtained by cropping a single mitochondrion with white padding. Using this approach, we computed the mean and standard deviation across 100 randomly selected prompts from the training set. To avoid incomplete or ambiguous examples, we excluded: objects smaller than $300px$ in area, and objects touching image borders.
For segmentation, we applied a fixed threshold of 0.3, as specified in the original CLIPSeg implementation.

Scribble prompts. To fairly compare DINOSim with Convpaint while minimizing human bias, we generated 10 randomized scribble prompts. Each prompt was created by:

1. Selecting a random image from the training set,
2. Using ground truth masks to draw three random straight lines over each class (foreground and background).

For background scribbles, we used a circular brush of 10 pixel radius, while foreground scribbles used a 30 pixel radius. Additional constraints were applied:

- Scribbles could not cross class boundaries or touch unconnected objects,
- Border regions were excluded to prevent ambiguity,
- Maximum scribble length was set to 300 px.

For a fair feature comparison, Convpaint was tested using DINOv2 features, as it performed best among VGG, DINOv2, and traditional image filters. Specifically, Convpaint used DINOv2 with ViT-S as the feature extractor.

Experimental Results

Datasets. We evaluated our method on two publicly available electron microscopy datasets for semantic segmentation. The first dataset, VNC (39) consists of a $4.7 \times 4.7 \times 1(\mu m)^3$ serial section transmission electron microscopy (ssTEM) volume of the *Drosophila melanogaster* third instar larva ventral nerve cord. Two volumes, each with $1024 \times 1024 \times 20$ voxels, were acquired, but only one was annotated. We split the labeled volume along the x-axis into two equal subsets ($512 \times 1024 \times 20$ voxels) for training and testing.
The second dataset, EPFL Hippocampus (40), captures a $5 \times 5 \times 5(\mu m)^3$ section of the CA1 region of a mouse hippocampus with focused ion beam scanning electron microscopy (FIB-SEM) at an isotropic resolution of 5 nm per voxel. The dataset contains a $2048 \times 1536 \times 1065$ voxel volume, mitochondria labeled in two sub-volumes (165 slices of 1024×768 pixels) for training and testing. We used an updated version of the annotations, *Lucchi++* (41), refined by neuroscientists and a senior biologist to correct misclassifications and boundary inconsistencies.
Since DINOv2 is a ViT trained with a patch size of 14×14 pixels and an image size of 518×518 pixels at the end of the training, we adopted a fixed size of 518×518 pixels ($H_p \times W_p$). For an input image of this size, DINOv2 produces an embedding tensor of $37 \times 37 \times \mathcal{F}$ ($\mathcal{H} \times \mathcal{W} \times \mathcal{F}$), where the feature dimension $\mathcal{F}$ varies depending on the model variant: 384 for ViT-S, 768 for ViT-B, 1024 for ViT-L, and 1536 for ViT-g.
Due to the 512-pixel with limitation of the VNC dataset, we cropped both datasets into tiles of 512×512 pixels ($H_t \times W_t$) to ensure consistency in processing.

Evaluation. We evaluate the performance of our method on semantic segmentation and object detection tasks.
For semantic segmentation, we assess results at the high-resolution original scale ($H \times W$). To do so, the upsampled predicted mask M from Eq. 5 is compared to the original ground truth annotations.
Semantic segmentation performance is quantified using the Intersection over Union (IoU) metric. Instead of computing the mean IoU across all classes, which is typically dominated by the background, we use the foreground IoU (IoUf), which exclusively measures segmentation accuracy for the foreground class. Specifically, we compute IoUf between the predicted and ground truth volumes:

- Each 2D slice of the volume is processed separately.
- The predicted masks from all slices are then combined to reconstruct the full 3D volume.

This volume-based evaluation aligns with previous research using the same datasets, ensuring comparability of results, even though our method does not explicitly process the third dimension.

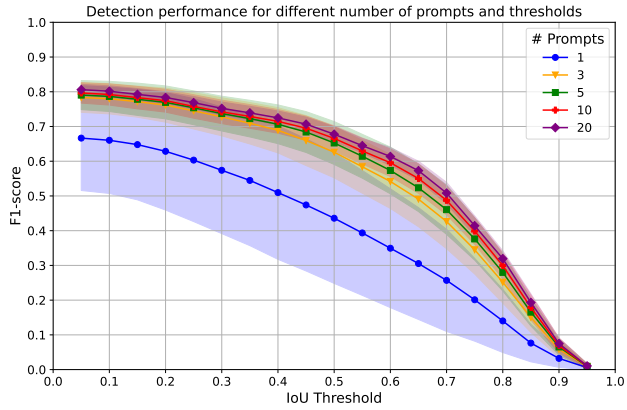

Fig. 2. F1-Score across different numbers of prompts and IoU thresholds. The plot illustrates the average F1-Score evolution across varying IoU threshold values. Different colors represent results for different numbers prompts. The standard deviation of each setting is indicated by a shaded region around the mean. This example is based on DINOv2 ViT-g using the VNC dataset, with 100 repetitions per experiment. Evaluated IoU thresholds range from 0.05 to 0.95 with a step size of 0.05.

For object detection, bounding boxes are extracted from the predicted mask M (Eq. 5) and compared to the bounding boxes computed from the ground truth mask. A bounding box is considered a true positive if its IoU with a ground truth box meets or exceeds a chosen threshold. Performance

is quantified using the F1-Score, accounting for both precision and recall. However, due to the low resolution of the embedding space, predicted bounding boxes often struggle to precisely separate closely positioned objects, leading to low IoU values when objects are in close proximity.
To address this limitation, we analyze the impact of the IoU threshold on true positive classification. As shown in Figure 2, the F1-score gradually decreases as the IoU threshold increases. This trend suggests that:

- Low IoU thresholds prevent excessive penalization of closely grouped objects, allowing detections that reasonably enclose multiple objects.
- Very permissive thresholds could lead to inflated true positive counts by allowing overly large, imprecise bounding boxes. If this were the case, we would expect a much sharper decline in F_1-score as the IoU threshold increases.

Based on these observations, we adopt a low IoU threshold of 0.05 for the object detection metrics. This threshold accounts for cases where multiple closely packed objects are enclosed within a single bounding box. Additionally, to reduce false positives, we exclude detections with an area below 300 pixels, as these small predictions often correspond to spurious or noise-induced artifacts.

DINOSim prompt selection. To evaluate the performance of DINOSim, prompts were selected from the training dataset and used as references to predict masks for the entire test dataset.
To automate the selection of object-related prompts, while simulating human prompting behavior, we developed a sampling strategy that prioritizes prompts located within object masks. Specifically, we computed a sampling probability map based on the ground truth segmentation masks, where:

- Background regions were assigned a probability of zero.
- Object borders had low sampling probability.
- Probability values increased towards the object's center, encouraging selection of points deep within the target structures.

This approach ensures that prompts are effectively placed within meaningful regions of interest.

Ablation studies.

Impact of k-NN the inference step. To evaluate whether using pseudo-labels and k-NN improves performance, we conducted an ablation study comparing results with and without the k-NN step. Specifically, we run the same experiments but replaced the k-NN computation from Eq. 4 with a direct distance measurement between E_{new} and the reference vector, as in Eq. 2.
After computing the distance, we clipped the values to the range $[min_{\forall r}(D'_r), max_{\forall r}(D'_r)]$, and applied normalization following Eq. 3, while keeping the post-processing step unchanged.
Results, summarized in Table 1, show a clear performance improvement when using k-NN and pseudo-labeling in DINOSim.

# of prompts	k-NN	IoUf	F1-Score
1	✗	0.343 ± 0.157	0.565 ± 0.171
	✓	**0.448 ± 0.166**	**0.662 ± 0.163**
3	✗	0.511 ± 0.097	0.682 ± 0.092
	✓	**0.614 ± 0.050**	**0.779 ± 0.039**

Table 1. Ablation study on the impact of k-NN. The table reports segmentation and object detection results over the VNC dataset, using different numbers of promptings. Results are presented with and without the k-NN step (using $k = 5$) in the inference stage of DINOSim. DINOv2 ViT-g embeddings were used for this experiment. Each value represents the mean ± standard deviation across 100 runs. The highest value for each number of prompts is shown in bold.

Effect of number of prompts and model size. We conducted additional ablation studies to examine how different number of prompts and model sizes contribute to the overall performance of DINOSim in object detection and segmentation. Specifically, we evaluated:

- Varying numbers of prompts (1, 3, 5, 10, and 20).
- Different DINOv2 model sizes (ViT-S, ViT-B, ViT-L, and ViT-g) trained with registers (42).

To avoid possible bias towards a specific dataset, experiments were performed in both VNC and Lucchi++ dataset, with each configuration repeated 100 times to ensure reliable results.
Object detection results, summarized in Figure 3, indicate that increasing the number of prompts improves F1-score stability across both datasets. However, no clear trend emerges. In the VNC dataset, performance generally improves with a higher number of prompts, whereas in the Lucchi++ dataset, only the two largest models benefit from additional prompts, while the smaller models exhibit a decline in performance. This suggests that the effect of prompt quantity is both model-dependent and dataset-specific. Nonetheless, performance quickly converges, aligning with the observations in Figure 2, where results obtained with three or five prompts are nearly indistinguishable. Regarding model size, larger models demonstrate slightly better and more stable performance.
Since object detection performance is derived from segmentation accuracy, similar trends were observed in both metrics (further details in Supplementary Material 3).
Although increasing the number of prompts and using larger models may increase runtime, DINOSim leverages GPU acceleration to mitigate computational overhead (see execution time analysis in Supplementary Material 4).
For all experiments, we set the segmentation threshold $\tau = 0.5$ as the default. However, slight adjustments to τ can further improve performance is specific scenarios (see threshold sensitivity analysis in Supplementary Material 1.
An example of the outputs of DINOSim is presented in Figure 4.

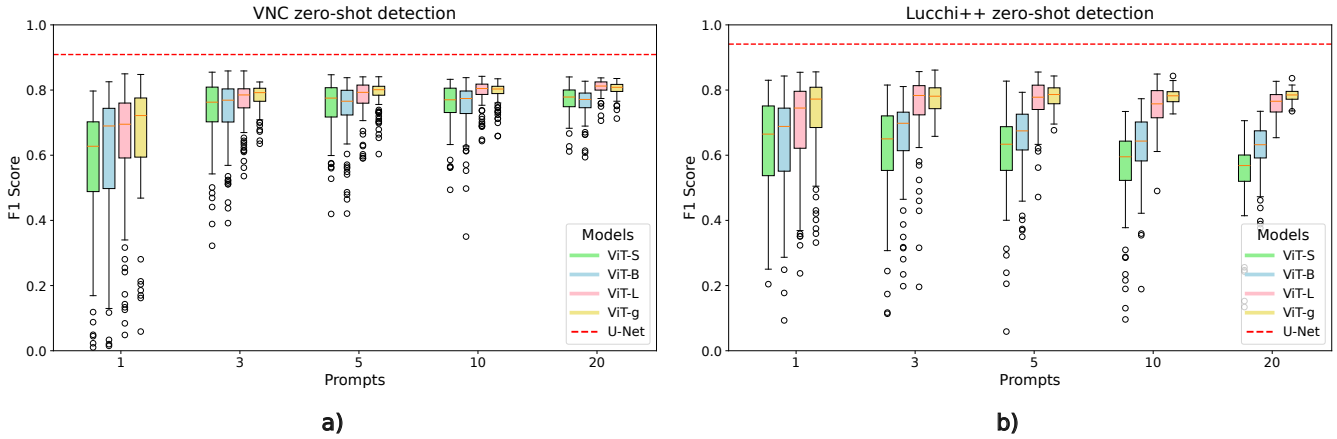

Fig. 3. Zero-shot object detection performance for different numbers of prompts. The figure presents the F1-score obtained with DINOSim in the test set of the VNC dataset in a) and Lucchi++ dataset in b), using different numbers of prompts, while the colors indicate different DINOv2 model sizes. The red line indicates the upper bound, corresponding to the best fully supervised result using the U-Net model. Each box represents 100 runs with random prompts.

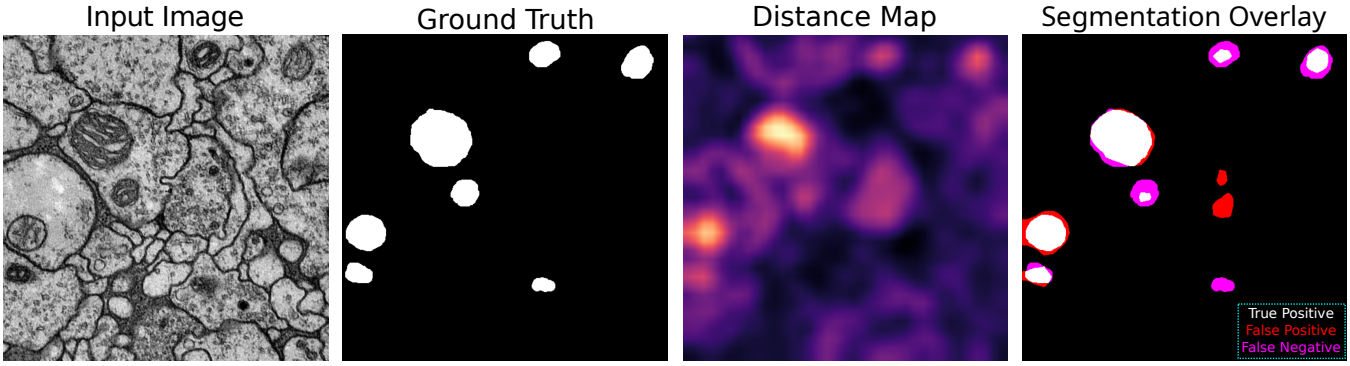

Fig. 4. Example of zero-shot segmentation result using DINOSim. The columns represent: (1) the original image from the test set of the VNC dataset, (2) the ground truth mask, (3) the upscaled distance map (lighter regions indicate higher similarity to the reference), and (4) the final predicted mask. In the predicted mask, true positives are shown in white, false positives in red, false negatives in magenta and true negatives in black. This example was generated using DINOv2 ViT-g and 3 random prompts from the train set of the VNC dataset.

Comparison with supervised and few-shot methods. As shown in Table 2 and Table 3, DINOSim with three prompts achieves zero-shot performance comparable to fully supervised methods at the embedding resolution.

Supervised methods. Focusing on the fully supervised results that work at embedding resolution (top rows of Table 2), k-NN performs best, establishing the upper bound for vector classification performance. This confirms that DINOv2 embeddings effectively cluster similar objects. However, working at embedding resolution significantly impacts segmentation metrics, placing all vector classification methods well behind the U-Net, which establishes the upper bound for pixel-wise segmentation.

In Table 3, we present object detection results. Apart from U-Net, which achieves highest scores, k-NN and MLP perform best, showing a balance between precision and recall, leading to higher F1-Score. However, their performance does not significantly differentiate them from the other models.

Few-shot and zero-shot methods. Among the few-shot or zero-shot baselines, CLIPSeg with text prompts (CLIPSeg-Txt) produced more accurate segmentations for the detected instances than image prompts (CLIP-Img, Table 2). However, the model misses a substantial number of mitochon-

Model	IoUf	
	VNC	Lucchi++
DT	0.546 ± 0.007	0.496 ± 0.003
RF	0.508 ± 0.002	0.404 ± 0.001
NB	0.610	0.467
k-NN	0.668	0.596
MLP	0.624 ± 0.003	0.586 ± 0.005
U-Net (35)	**0.839 ± 0.008**	**0.895 ± 0.005**
CLIPSeg-Img (26)	0.106 ± 0.034	0.106 ± 0.027
Convpaint (33)	0.392 ± 0.098	0.240 ± 0.078
CLIPSeg-Txt (26)	0.148	0.122
DINOSim 1P (ours)	0.448 ± 0.166	0.421 ± 0.119
DINOSim 3P (ours)	**0.614 ± 0.050**	**0.486 ± 0.062**

Table 2. Semantic segmentation performance of fully supervised and few-shot methods. The table is divided into two sections by a double-line: The first section contains semantic segmentation results from fully supervised models with vector classification methods (white) and models trained with original images (light-gray). The second section includes all few-shot approaches, including one-shot (gray) and zero-shot (dark-gray) methods. For DINOSim, we report results using one prompt (1P) and three prompts (3P), both using DINOv2 ViT-g. Each value represents the mean ± standard deviation across 10 runs, except for DINOSim and CLIP-seg-Img (100 runs) and CLIPSeg-Txt (1 run, using the best prompt found). The highest value in each section is shown in bold.

Model	VNC			Lucchi++		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
DT	0.884 ± 0.020	0.740 ± 0.011	0.805 ± 0.010	0.901 ± 0.004	0.800 ± 0.005	0.848 ± 0.005
RF	**0.974 ± 0.007**	0.640 ± 0.006	0.773 ± 0.006	**0.969 ± 0.002**	0.724 ± 0.002	0.828 ± 0.001
NB	0.677	**0.943**	0.788	0.627	0.918	0.745
k-NN	0.916	0.793	0.850	0.912	0.863	0.887
MLP	0.901 ± 0.007	0.782 ± 0.006	0.838 ± 0.004	0.912 ± 0.003	0.866 ± 0.003	0.888 ± 0.003
U-Net (35)	0.903 ± 0.012	0.915 ± 0.014	**0.909 ± 0.008**	0.928 ± 0.009	**0.954 ± 0.004**	**0.941 ± 0.005**
CLIPSeg-Img (26)	0.499 ± 0.124	0.297 ± 0.078	0.367 ± 0.088	0.583 ± 0.089	0.351 ± 0.109	0.430 ± 0.102
Convpaint (33)	0.521 ± 0.098	**0.794 ± 0.099**	0.605 ± 0.104	0.363 ± 0.109	0.657 ± 0.274	0.432 ± 0.157
Grounding DINO (25)	0.286	0.021	0.039	0.292	0.173	0.217
CLIPSeg-Txt (26)	**0.927**	0.197	0.325	**0.715**	0.302	0.424
DINOSim 1P (ours)	0.800 ± 0.182	0.610 ± 0.192	0.662 ± 0.163	0.686 ± 0.125	0.787 ± 0.165	0.722 ± 0.120
DINOSim 3P (ours)	0.791 ± 0.085	0.781 ± 0.068	**0.779 ± 0.039**	0.681 ± 0.074	**0.906 ± 0.027**	**0.774 ± 0.042**

Table 3. Object detection performance of fully supervised and few-shot methods. The table is divided into two sections by a double-line: The first section contains object detection results from fully supervised models with vector classification methods (white) and models trained with original images (light-gray). The second section includes all few-shot approaches, including one-shot (gray) and zero-shot (dark-gray) methods. For DINOSim, we report results using one prompt (1P) and three prompts (3P), both using DINOv2 ViT-g. Each value represents the mean ± standard deviation across 10 runs, except for DINOSim and CLIPSeg-Img (100 runs) and CLIPSeg-Txt and Grounding DINO (1 run, using the best prompt found). The highest value in each section is shown in bold.

dria, leading to severe under-segmentation. This resulted in a lower IoUf score, despite a high detection precision of 0.927 in VNC and 0.715 in Lucchi++. With image prompts, CLIPSeg-Img tended to oversegment, capturing excessively large regions that encompassed multiple objects (see Figure 5). This led to similar F1-scores, yet with low segmentation scores in both datasets. A qualitative comparison of segmentation results from the few-shot methods is provided in Figure 5.

Similarly, Grounding DINO detected only a small subset of mitochondria, resulting in very low detection metrics. In contrast, ConvPaint demonstrated slightly improved segmentation performance compared to CLIPSeg across both datasets. It outperformed both CLIPSeg and Grounding DINO on the VNC dataset but achieved a comparable F1-score to CLIPSeg on Lucchi++. Notably, ConvPaint exhibited a tendency to oversegment (see Figure 5 and Table 3), as indicated by its higher recall than precision.

Generalization across datasets. We evaluated DINOSim's robustness to dataset shifts by testing its performance when prompted and evaluated on different datasets. To reduce extreme variance, we used three references (3P) instead of just one (1P), and the results are summarized in Table 4.

Although both datasets (VNC and Lucchi++) focus on mitochondria segmentation, they differ in several key aspects: they originate from different species (fruit fly vs. mouse), they were acquired using different electron microscopy techniques, and they present different structures and contrast distributions. These differences introduce domain shifts, which pose a significant challenge for state-of-the-art segmentation methods (43). For instance, a U-Net-like model trained on VNC achieves an IoUf score of 0.351 ± 0.101 on Lucchi++, but fails completely (0.009 ± 0.010 IoUf scores) when trained on Lucchi++ and tested on VNC.

In contrast, DINOSim maintains consistent performance across datasets, demonstrating generalization capabilities in both segmentation and detection tasks. Unlike fully supervised methods, which degrade under domain shifts, DINOSim remains robust to changes in dataset characteristics, exceeding the generalization ability of standard deep learning approaches.

Dataset		DINOSim 3P (ViT-g)	
Reference	Inference	IoUf	F1-Score
VNC	VNC	0.614 ± 0.050	0.779 ± 0.039
Lucchi++	VNC	0.563 ± 0.049	0.756 ± 0.043
VNC	Lucchi++	0.455 ± 0.056	0.769 ± 0.038
Lucchi++	Lucchi++	0.486 ± 0.062	0.774 ± 0.042

Table 4. Cross-dataset semantic segmentation and object detection performance. The first column indicates the dataset used for reference selection, while the second column indicates the dataset used for inference. The table provides segmentation IoUf and detection F1-Score. Each value represents the mean ± standard deviation over 100 runs, using DINOv2 ViT-g.

Discussion

Defining the zero-shot boundary. Fairly comparing different zero-shot approaches is inherently challenging, as each method relies on distinct prompting strategies. Additionally, the boundary between zero-shot and one-shot learning is often ambiguous.

In image-based prompting, segmentation or recognition a specific object can be ambiguous when multiple objects are present in the image. One strategy to address this is to create object-centric prompts by cropping around the object of interest. However, this technique can be interpreted as leveraging detection or segmentation labels, making it closer to a one-shot approach. From a user's perspective, providing a textual prompt versus selecting a bounding box requires nearly the same level of effort. Moreover, when prompt engineering is required, the need for multiple refinements often makes bounding box prompting a faster alternative than the textual zero-shot approach, as it reduces human interactions.

Interactive learning methods further complicate this distinction. While they do not rely on fully labeled datasets, they

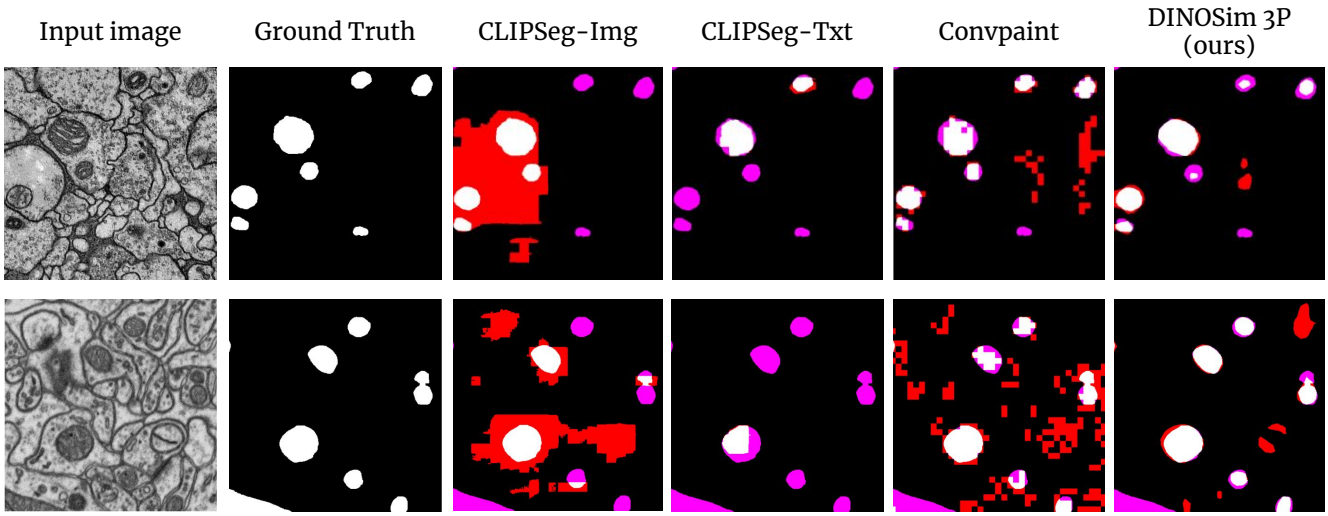

Fig. 5. Qualitative comparison of few-shot segmentation methods. From left to right: the original image, the ground truth mask, the predicted mask using CLIPSeg with image prompts, the predicted mask using CLIPSeg with textual prompts, the predicted mask using Convpaint, the predicted mask using DINOSim (ViT-g, 3 prompts). In the predicted masks, true positives are in white, false positives in red, false negatives in magenta and true negatives in black. The first row presents an example from the VNC dataset, while the second row presents an example from the Lucchi++ dataset.

exist in a gray area between zero-shot and few-shot learning. However, as the number of iterations increases, interactive learning demands more training time and labeling effort than standard few-shot approaches. Additionally, these methods require highly specific human annotations to correct the model's predictions.

Paradoxically, this can make interactive learning more label-intensive than other few-shot techniques. Another challenge is that interactive models can converge on suboptimal labels, making it difficult to decide when to stop iterating. This uncertainty can lead to excessive relabeling and retraining, increasing annotation effort.

Limitations of DINOSim. DINOSim segments objects based on semantic similarity in the embedding space of a fixed, pretrained image encoder. This reliance on precomputed embeddings limits control over what the model considers "similar". For instance, if the goal is to differentiate damaged and non-damaged mitochondria, DINOSim may struggle to distinguish them, as both could be represented similarly in the embedding space. As a result, both categories might be segmented together, reducing its effectiveness for fine-grained classification tasks.

Conclusions

In this work, we addressed the challenge of limited labeled data in electron microscopy images using few-shot learning. We introduced DINOSim, a zero-shot object detection and semantic segmentation method that leverages DINOv2 embeddings without requiring additional training or label data. Compared to existing few-shot approaches, DINOSim achieved superior performance in object detection and semantic segmentation with fewer prompts.

One key limitation observed is the reduced effectiveness of working directly with embeddings for pixel-level segmentation tasks. However, inspired by the idea of Grounded SAM (11), DINOSim could be integrated with SAM (9, 10) to enhance segmentation precision. While this combination may improve results, it would come at the cost of increased computational overhead and reduced efficiency.

Future research directions include exploring multi-resolution techniques or super-resolution in the embedding space (44), to improve pixel-level segmentation performance, and refining object detection by incorporating additional instance or boundary-aware, information.

Despite these challenges, DINOSim offers a simple, fast, and accessible zero-shot object detection and semantic segmentation solution across diverse environments and object types. To facilitate its adoption, we have released an open-source Napari plugin (https://github.com/AAitorG/napari-dinoSim), enabling easy implementation and broader usability within the bioimage analysis community.

ACKNOWLEDGEMENTS

This work is partially supported by grant GIU23/022 (to I.A-C.) funded by the University of the Basque Country (UPV/EHU), and grant PID2021-126701OB-I00 (to I.A-C.), funded by the Ministerio de Ciencia, Innovación y Universidades, AEI, MCIN/AEI/10.13039/501100011033, and by "ERDF A way of making Europe". E.G-de-M. acknowledges the support of the Gulbenkian Foundation (Fundação Calouste Gulbenkian), the European Union through the Horizon Europe program (AI4LIFE project with grant agreement 101057970-AI4LIFE granted to the Optical Cell Biology laboratory at Instituto Gulbenkian de Ciência), the European Molecular Biology Organization (EMBO) Postdoctoral Fellowship (EMBO ALTF 174-2022 to E.G-de-M.) and the Fundação para a Ciência e Tecnologia, Portugal (FCT fellowship 2023.09182.CEECIND to E.G-de-M.). Funded by the European Union. However, the views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Bibliography

1. Simone Antonelli, Danilo Avola, Luigi Cinque, Donato Crisostomi, Gian Luca Foresti, Fabio Galasso, Marco Raoul Marini, Alessio Mecca, and Daniele Pannone. Few-shot object detection: A survey. *ACM Comput. Surv.*, 54(11s), sep 2022. ISSN 0360-0300. doi: 10.1145/3519022.
2. Wenqi Ren, Yang Tang, Qiyu Sun, Chaoqiang Zhao, and Qing-Long Han. Visual semantic segmentation based on few/zero-shot learning: An overview. *IEEE/CAA Journal of Automatica Sinica*, 11(5):1106–1126, 2024. doi: 10.1109/JAS.2023.123207.

3. Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022.
4. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
5. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.
6. Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning, 2023.
7. Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, June 2022.
8. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.
9. Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, October 2023.
10. Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024.
11. Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.
12. Uriah Israel, Markus Marks, Rohit Dilip, Qilin Li, Morgan Schwartz, Elora Pradhan, Edward Pao, Shenyi Li, Alexander Pearson-Goulart, Pietro Perona, Georgia Gkioxari, Ross Barnowski, Yisong Yue, and David Van Valen. A foundation model for cell segmentation, 2023.
13. Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, October 2021.
14. Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.
15. Joana Palés Huix, Adithya Raju Ganeshan, Johan Fredin Haslum, Magnus Söderberg, Christos Matsoukas, and Kevin Smith. Are natural domain foundation models useful for medical image classification? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7634–7643, January 2024.
16. Chongyu Qu, Tiezheng Zhang, Hualin Qiao, Jie Liu, Yucheng Tang, Alan Yuille, and Zongwei Zhou. Abdomenatlas-8k: Annotating 8,000 ct volumes for multi-organ segmentation in three weeks, 2023.
17. Melissa Linkert, Curtis T. Rueden, Chris Allan, Jean-Marie Burel, Will Moore, Andrew Patterson, Brian Loranger, Josh Moore, Carlos Neves, Donald MacDonald, Aleksandra Tarkowska, Caitlin Sticco, Emma Hill, Mike Rossner, Kevin W. Eliceiri, and Jason R. Swedlow. Metadata matters: access to image data in the real world. *Journal of Cell Biology*, 189 (5):777–782, 05 2010. ISSN 0021-9525. doi: 10.1083/jcb.201004104.
18. Ryan Conrad and Kedar Narayan. Instance segmentation of mitochondria in electron microscopy images with a generalist deep learning model trained on a diverse dataset. *Cell Systems*, 14(1):58–71.e5, Jan 2023. ISSN 2405-4712. doi: 10.1016/j.cels.2022.12.006.
19. Jiaxing Huang, Kai Jiang, Jingyi Zhang, Han Qiu, Lewei Lu, Shijian Lu, and Eric Xing. Learning to prompt segment anything models, 2024.
20. Zabir Al Nazi and Wei Peng. Large language models in healthcare and medical domain: A review. *Informatics*, 11(3), 2024. ISSN 2227-9709. doi: 10.3390/informatics11030057.
21. Md Adnan Arefeen, Biplob Debnath, and Srimat Chakradhar. Leancontext: Cost-efficient domain-specific question answering using llms. *Natural Language Processing Journal*, 7: 100065, 2024. ISSN 2949-7191. doi: https://doi.org/10.1016/j.nlp.2024.100065.
22. Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. Towards expert-level medical question answering with large language models, 2023.
23. Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, Jan 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-44824-z.
24. Anwai Archit, Sushmita Nair, Nabeel Khalid, Paul Hilt, Vikas Rajashekar, Marei Freitag, Sagnik Gupta, Andreas Dengel, Sheraz Ahmed, and Constantin Pape. Segment anything for microscopy. *bioRxiv*, 2023. doi: 10.1101/2023.08.21.554208.
25. Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024.
26. Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7086–7096, June 2022.
27. Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.
28. Xin Rong. word2vec parameter learning explained, 2016.
29. Maxime Bucher, Tuan-Hung VU, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
30. Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, 1995. doi: 10.1109/ICDAR.1995.598994.
31. Ignacio Arganda-Carreras, Verena Kaynig, Curtis Rueden, Kevin W Eliceiri, Johannes Schindelin, Albert Cardona, and H Sebastian Seung. Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics*, 33(15):2424–2426, 03 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx180.
32. Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N Straehle, Bernhard X Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, Kemal Eren, Jaime I Cervantes, Buote Xu, Fynn Beuttenmueller, Adrian Wolny, Chong Zhang, Ullrich Koethe, Fred A Hamprecht, and Anna Kreshuk. ilastik: interactive machine learning for (bio)image analysis. *Nature Methods*, 16(12):1226–1232, December 2019.
33. Lucien Hinderling, Guillaume Witz, Roman Schwob, Ana Stojiljković, Maciej Dobrzyński, Mykhailo Vladymyrov, Joël Frei, Benjamin Grädel, Agne Frismantiene, and Olivier Pertz. Convpaint - interactive pixel classification using pretrained neural networks. *bioRxiv*, 2024. doi: 10.1101/2024.09.12.610926.
34. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.
35. Daniel Franco-Barranco, Arrate Muñoz-Barrutia, and Ignacio Arganda-Carreras. Stable deep neural network architectures for mitochondria segmentation on electron microscopy volumes. *Neuroinformatics*, 20(2):437–450, Apr 2022. ISSN 1559-0089. doi: 10.1007/s12021-021-09556-1.
36. Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
37. Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.
38. Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2018.
39. Stephan Gerhard, Jan Funke, Julien Martel, Albert Cardona, and Richard Fetter. Segmented anisotropic sstem dataset of neural tissue. *figshare*, pages 0–0, 2013.
40. Aurélien Lucchi, Kevin Smith, Radhakrishna Achanta, Graham Knott, and Pascal Fua. Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features. *IEEE Transactions on Medical Imaging*, 31(2):474–486, 2012. doi: 10.1109/TMI.2011.2171705.
41. Vincent Casser, Kai Kang, Hanspeter Pfister, and Daniel Haehn. Fast mitochondria detection for connectomics. In Tal Arbel, Ismail Ben Ayed, Marleen de Bruijne, Maxime Descoteaux, Herve Lombaert, and Christopher Pal, editors, *Proceedings of the Third Conference on Medical Imaging with Deep Learning*, volume 121 of *Proceedings of Machine Learning Research*, pages 111–120. PMLR, 06–08 Jul 2020.
42. Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2024.
43. Daniel Franco-Barranco, Julio Pastor-Tronch, Aitor González-Marfil, Arrate Muñoz-Barrutia, and Ignacio Arganda-Carreras. Deep learning based domain adaptation for mitochondria segmentation on em volumes. *Computer Methods and Programs in Biomedicine*, 222: 106949, 2022. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2022.106949.
44. Stephanie Fu, Mark Hamilton, Laura Brandt, Axel Feldman, Zhoutong Zhang, and William T. Freeman. Featup: A model-agnostic framework for features at any resolution, 2024.

Supplementary Note 1: Analysis of the distance map threshold values

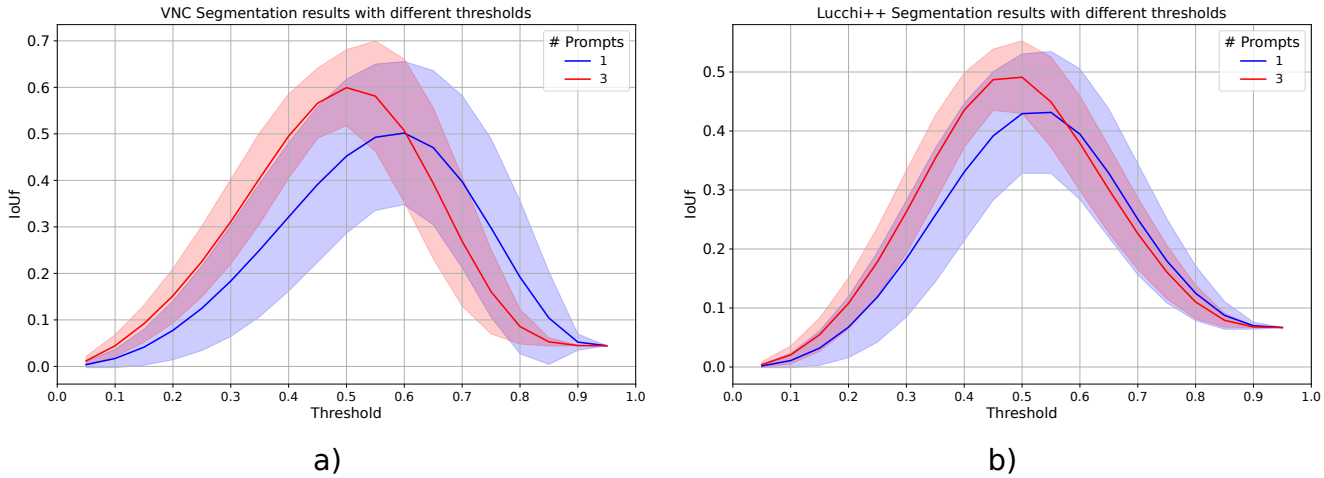

Fig. 6. Semantic segmentation performance at different thresholds. The figure shows the IoUf as a function of the threshold values used to extract masks from the predicted distance maps of the VNC test dataset in a) and Lucchi++ test dataset in b), using DINOSim with the ViT-g. The blue line represents the average result from 100 runs using the three prompts, and the shaded blue area shows the standard deviation in these results.

We realized that tuning the threshold used to extract masks out of distance maps, led to better results, but whether moving higher or lower, depends on the data. If we perform a posterior analysis using the ground truth, we can evaluate the performance on segmentation using different thresholds, as we can see in Figure 6. In both datasets with 3 prompts, using 0.5 threshold seems to be the best option. If we evaluate the performance evolution with 1 prompt, that not only the variance is higher, but also the best average IoUf is found with different thresholds, 0.6 in case of VNC and 0.55 in case of Lucchi++. As the best threshold tends to be close to 0.5, we decided to make this the default threshold.

Supplementary Note 2: Impact of the number of neighbors in k-NN

We evaluated the influence of different number of neighbors in the segmentation and detection performance, in both fully supervised and DINOSim approach, shown in Table 5 respectively. In both scenarios, results are very similar, showing a strong invariance to the number of neighbors. With slightly better results, we selected k=5 for the fully supervised method. While in DINOSim, slightly better results appears with more neighbors, this could be counterproductive. As the method works with generated pseudo-labels, may happen to prompted class to belong to a small object, with not many positives. To deal with this unbalance scenario, by default we decided to use a lower amount of neighbors k=5.

k	IoUf	F1-Score
1	0.6514	0.8746
3	**0.6684**	0.8889
5	0.6677	**0.8915**
7	0.6655	0.8757
9	0.6617	0.8724

(a) Fully supervised results (ViT-g)

k	IoUf	F1-Score
1	0.584 ± 0.063	0.753 ± 0.055
3	0.600 ± 0.060	0.768 ± 0.048
5	0.605 ± 0.058	0.774 ± 0.046
7	0.607 ± 0.058	0.776 ± 0.044
9	**0.609 ± 0.057**	**0.778 ± 0.042**

(b) DINOSim 3P results (ViT-g)

Table 5. Impact of the number of neighbors in k-NN. Semantic segmentation and object detection results using a different number of neighbors in the k-NN step for the fully supervised (a) and DINOSim (b) approaches.

Supplementary Note 3: Impact of the number of prompts and the model size in semantic segmentation results

In this section, variation in segmentation performance for different amount of prompts and different model sizes are evaluated. Regarding the model size, there is a clear tendency of larger models offering better, and more stable segmentation results, see Figure 7. Notably, there is a gap between the two smallest and largest models, the larger the number of prompts the larger the gap. The same conclusions can be seen in detection (see figure 3).

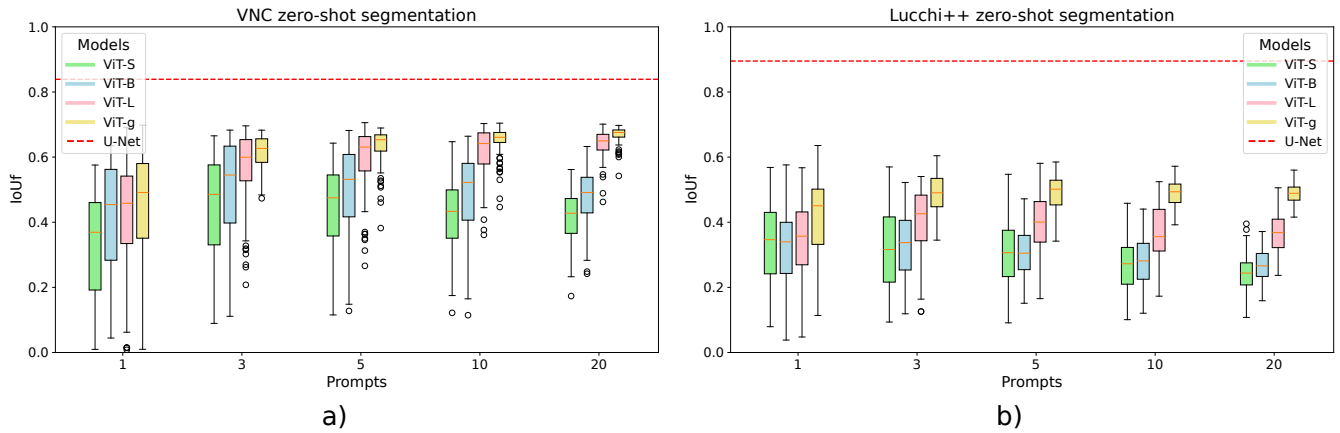

Fig. 7. Zero-shot semantic segmentation performance for different prompts. The figure presents the IoUf obtained with DINOSim in the test set of the VNC dataset in a) and Lucchi++ dataset in b), using different number of prompts, while the colors indicate various DINOv2 model sizes. The red line marks the upper bound, showing the best fully supervised result using the U-Net model. Each box represents 100 runs with random prompts.

Supplementary Note 4: Execution times

All experiments were conducted on a system featuring an Intel Xeon Silver 4210R CPU and an NVIDIA GeForce RTX 3090 GPU with CUDA 12.2. The implementation was built using PyTorch version 2.4.1. Runtime performance details are summarized in Table 6, using the largest model, DINOv2 ViT-g. Since image embeddings are computed only once and can be reused for any number of subsequent prompts, this design ensures that DINOSim operates with improved speed and responsiveness.

Model	Prompts	CPU (ms)	GPU (ms)
ViT-S		16.0 ± 1.6	4.7 ± 0.3
ViT-B	3	24.8 ± 3.1	5.1 ± 0.3
ViT-L		30.6 ± 4.2	5.0 ± 0.2
	1	26.8 ± 0.8	4.7 ± 0.1
	3	41.7 ± 6.2	5.6 ± 0.2
ViT-g	5	85.7 ± 9.6	6.4 ± 0.2
	10	149.6 ± 12.0	8.2 ± 0.5
	20	255.0 ± 18.4	11.1 ± 0.6

(a) DINOSim running times.

Model	CPU (ms)	GPU (ms)
ViT-g	10,733.3 ± 129.8	272.3 ± 16.6
ViT-L	3,224.5 ± 187.0	91.5 ± 16.2
ViT-B	1,154.7 ± 65.6	32.9 ± 16.4
ViT-S	396.2 ± 41.1	12.1 ± 0.9

(b) DINOv2 embedding calculation times.

Table 6. Execution times. a) The table presents, by column: DINOv2 model size, the number of prompts, runtimes on the CPU, and runtimes on the GPU. The reported times reflect the duration required to generate predicted masks using precomputed image embeddings. For CPU runtimes, values were averaged over 400 repetitions, while GPU runtimes were averaged over 4000 repetitions with varying random prompts. b) The table presents, by column: DINOv2 model size, runtimes on the CPU, and runtimes on the GPU. The reported times reflect the duration required to obtain image embeddings using DINOv2. Times were averaged over 40 runs. All times in a) and b) are measured per crop of 518×518 pixels.