Unified Multi-Task Foundation Model for *C. elegans*

Abstract

Recent advances in deep learning have transformed biological image analysis, yet most existing models are task-specific and computationally intensive, limiting their scalability in real-world laboratory workflows. Here, we propose a unified and lightweight foundation model tailored for *Caenorhabditis elegans*, capable of performing multiple vision tasks including semantic and instance segmentation, object detection, keypoint and arbitrary-point tracking, brightfield and fluorescence denoising, and super-resolution, all within a shared backbone architecture. The model is trained on a comprehensive dataset integrating public and in-house imaging data, covering diverse imaging modalities and biological contexts. To evaluate biological relevance and generalization, we design three benchmark applications: drug screening, transgenic phenotyping, and behavior–neural activity coupling. Our approach emphasizes high performance, efficiency, and deployability, enabling real-time analysis in high-throughput experimental settings. This work establishes a modular and scalable foundation for multi-task visual inference in *C. elegans*, offering a broadly applicable framework for lightweight foundation models in biological imaging.

Background and Motivation

Caenorhabditis elegans is a premier model organism in biology, used extensively for studies in genetics, neurobiology, aging, and drug discovery[1][2]. Its transparent body and ease of cultivation make it ideal for high-content imaging and behavioral analysis[3][4]. However, extracting quantitative information from *C. elegans* imaging data remains a significant bottleneck[5]. A variety of computer vision (CV) tasks are needed – including worm segmentation, detecting individuals, tracking their movements, enhancing image resolution, and denoising – yet current approaches treat these tasks in isolation. This fragmentation of *C. elegans* imaging tools impedes efficient analysis and reproducibility.

Traditionally, laboratories have relied on separate, task-specific tools: for example, threshold-based segmentation and skeletonization pipelines for morphology, custom trackers for locomotion (e.g. Multi-Worm Tracker, Track-A-Worm), and specialized algorithms for fluorescence signal extraction[6][7]. Early systems like the Multi-Worm Tracker could follow multiple worms via classical vision methods[6], but they struggled under challenging conditions (occlusions, variable lighting)[8]. Indeed, conventional segmentation algorithms often fail on noisy or crowded images, forcing researchers to discard frames where worms overlap or coil[9][10]. As a result, large portions of worm behavior – especially group interactions or complex postures – have been impossible to quantify with existing end-to-end

methods[9]. This leads to a **fragmented toolkit**, where each aspect of worm phenotyping (e.g. locomotion vs. morphology) requires a different model or manual intervention.

Modern deep learning techniques are beginning to surmount these limitations, but they too have emerged in a piecemeal fashion. For instance, Liu *et al.* (2025) introduced a YOLOv8+ByteTrack-based framework for worm **detection and multi-object tracking** that achieves remarkable accuracy (~99.5% precision, 153 FPS) in real time[11]. Separately, Deserno and Bozek (2023) developed **WormSwin**, a transformer-based model for **instance segmentation** of worms, capable of resolving overlapping individuals with 99% average precision[12]. Other groups have proposed models for specific tasks like **skeleton extraction**[13] or **cell/nuclear segmentation in worms**[14]. While each of these advances is important, they remain isolated solutions targeting single problems. A researcher wanting to fully characterize worm phenotypes might need to run several disconnected pipelines – one for segmentation, another for tracking, another for image restoration – each with its own data requirements and potential inconsistencies. This not only increases computational and labor overhead, but also risks compounding errors (e.g. a tracking algorithm failing due to poor segmentation input).

These challenges point to a clear need for **integrated, multi-task approaches** in *C. elegans* image analysis. In the broader biomedical imaging field, there is a growing recognition that **holistic analysis** – jointly tackling interdependent subtasks like segmentation, detection, and recognition – can outperform siloed methods[15]. A recent Nature Methods study introduced *BiomedParse*, a foundation model that **jointly learns** to segment, detect, and classify biomedical objects across diverse imaging modalities[15]. Their joint-learning approach not only improved accuracy on individual tasks but also enabled new capabilities, such as segmenting all objects described by a text prompt[16][17]. This success exemplifies the power of **foundation models** in biomedicine: large-scale models that learn versatile representations applicable to many tasks and domains[18]. Likewise, in the *C. elegans* domain, Guisnet & Hendricks (2025) demonstrated a compound AI system (*TWARDIS*) that leverages general vision transformers (the Segment Anything Model, SAM) for worm segmentation and posture analysis across brightfield and fluorescence data[19][10]. *TWARDIS* showed that with foundation-model-level segmentation (via SAM), worms can be segmented even in noisy, low-contrast images without frame rejection, achieving near-perfect agreement with manual outlines[10]. It further handled multi-worm videos and neuron imaging by applying the same robust vision backbone across modalities[10][20]. These developments underscore two key insights: (1) **Multi-task and multi-modal modeling** can dramatically enhance performance and generality[15][10], and (2) **Foundation models** (pre-trained on large data) can serve as powerful backbones for biomedical image analysis[19][18].

Despite this promise, current solutions also highlight limitations that our proposal aims to address. Most existing foundation models (like SAM or BiomedParse) are extremely large and computationally intensive, hindering their deployment in typical lab settings or real-time experiments[21][22]. For example, SAM's ViT-H backbone provides excellent segmentation quality but requires GPU-heavy resources, spurring efforts like MobileSAM to create lighter versions[23]. MobileSAM replaced SAM's heavy encoder with a Tiny-ViT, significantly reducing model size and speeding up inference while maintaining competitive performance[23]. However, even MobileSAM sacrificed some fine-grained accuracy, especially on specialized tasks (e.g. subtle medical image details)[23][24]. More broadly, current biomedical foundation models tend to focus on *a subset* of tasks (often segmentation and classification/detection)[15]. None so far integrate *temporal* tasks like object tracking or diverse image-to-image tasks (super-resolution, denoising) into one framework. In *C. elegans* research, this means that although we can now apply large vision models to segmentation[19] or tracking[11], we still lack a **unified model** that handles **all the necessary analyses concurrently**. The consequence is that laboratories must juggle multiple models and possibly retrain each for different imaging conditions (for instance, a model trained on brightfield images might falter on fluorescence or high-magnification microscopy, and vice versa).

In summary, there is both a strong motivation and a timely opportunity to develop a **multi-task, multi-modal foundation model** specifically for *C. elegans*. Such a model would unify semantic and instance segmentation, object detection, multi-object tracking, super-resolution reconstruction, image denoising, and even user-guided point tracking within **one lightweight deployable framework**. By sharing a common backbone, the model could exploit synergies between tasks – e.g. features that help identify a worm might also aid denoising and vice versa – to improve overall robustness[15][25]. Importantly, by employing knowledge distillation and efficient design, we aim to make the model **deployable in real-world settings** (e.g. running at the microscope or on standard lab computers), bridging the gap between cutting-edge AI and everyday biological research. This proposal directly addresses the fragmentation and technical barriers in current *C. elegans* imaging methods, aligning with emerging trends in biomedical AI while focusing on the specific needs of worm researchers. Ultimately, our unified model would **remove critical bottlenecks** in *C. elegans* phenotyping, allowing scientists to focus on biology rather than wrestling with disparate image analysis tools[26].

Research Hypotheses and Questions

Building on the background above, we posit the following hypotheses and key questions for this project:

- **H1: Shared Backbone for Multi-Task Adaptability** – *Hypothesis:* A single deep neural network backbone can learn a rich representation of *C. elegans* images that is **simultaneously effective for multiple tasks** (segmentation,

detection, tracking, restoration, etc.), achieving performance on each task on par with or exceeding specialized single-task models. We hypothesize that multi-task training will *not* significantly compromise individual task accuracy; instead, certain tasks will **complement each other** (e.g. segmentation features improving detection and tracking)[15][25]. **Question:** Can one model truly excel at diverse tasks like segmentation and super-resolution at the same time, or will there be trade-offs (negative transfer)? How can we design the architecture to maximize positive transfer and minimize conflicts between tasks?

- **H2: Multi-Modal and Cross-Domain Generalization** – *Hypothesis:* Training on **multiple imaging modalities and experimental conditions** (brightfield vs. fluorescence imaging, low-magnification vs. high-magnification, different labs' datasets) using a unified backbone will yield **domain-agnostic features**, thereby enhancing the model's ability to generalize to new, unseen conditions. In other words, a multi-modal foundation model for worms will handle cross-domain shifts more gracefully than single-modality models. **Question:** Will a model trained on, say, brightfield videos and fluorescence microscopy data generalize to other modalities or lighting conditions without fine-tuning? To what extent does multi-modal training improve **robustness to variations** in noise, illumination, and imaging devices (e.g. different microscope setups)[10]? We aim to test whether jointly learning from diverse data (possibly with domain-specific augmentations or normalization) improves performance on each domain compared to separate training.

- **H3: Efficacy of Knowledge Distillation for Model Compression** – *Hypothesis:* We hypothesize that after training a large-capacity "teacher" model on all tasks, we can apply **knowledge distillation** to obtain a **lightweight "student" model** that retains most of the performance (within ~5-10% of accuracy metrics) while significantly reducing model size and inference time. This student model should be deployable on resource-limited hardware (e.g. standard CPUs or embedded GPUs) for real-time or high-throughput analysis. **Question:** Can distillation or model compression techniques produce a small model that **maintains high accuracy across all tasks**? What distillation strategy (e.g. layer-wise feature distillation, output logits distillation, or multi-stage teacher–student training) is most effective for a *multi-task* scenario? We will investigate if a student model can achieve comparable results to the teacher across segmentation quality, detection precision, tracking continuity, and image restoration fidelity[27]. Additionally, we ask: how does the performance vs. efficiency trade-off scale – e.g. is there an optimal student model size that balances speed and accuracy for worm analyses?

- **H4: Multi-Task Learning and Task Interdependence** – *Hypothesis:* The inclusion of certain tasks will benefit others (due to shared representations),

but some task combinations might introduce conflicts. For example, learning a denoising task might implicitly encourage the backbone to focus on fundamental structures (improving segmentation in noisy images), while learning an instance segmentation task might sharpen localization features that also help tracking. Conversely, some tasks (e.g. super-resolution reconstruction) might compete for network capacity without careful balancing. **Question:** Which tasks are **synergistic** when learned together, and are any tasks **antagonistic** (causing negative transfer)? We will test sub-hypotheses such as: Does joint training with a **denoising objective** improve segmentation accuracy under low signal-to-noise conditions compared to a model trained on segmentation alone? Does adding a **tracking objective** (temporal consistency) improve detection recall by leveraging motion cues? These questions will be addressed via ablation experiments where we enable/disable tasks during training and measure the impact on other task performances.

- **H5: Feasibility and Impact for Biological Research** – *Hypothesis:* By meeting the above technical goals, the resulting model will drastically streamline *C. elegans* imaging workflows, making advanced analyses accessible and standardized. We expect that our foundation model, applied to real experimental datasets, will **outperform existing analysis pipelines** in accuracy and throughput, and do so in a unified manner. **Question:** Can our model demonstrate tangible improvements in **real-world scenarios** such as drug screening assays or neural imaging experiments, compared to current best practices? We will evaluate the model's performance and utility in case-study applications (outlined below) to verify that it indeed removes analysis bottlenecks and yields biologically meaningful insights that were previously difficult to obtain.

By addressing these hypotheses, we aim to validate the core premise that a *multi-task, multi-modal foundation model* can be built for *C. elegans* and that it will offer significant advantages in adaptability, efficiency, and scientific impact over existing fragmented tools.

Proposed Methodology and Model Framework

To test the above hypotheses, we will design a **unified deep learning framework** comprising a shared backbone network and multiple task-specific "heads," followed by a model compression (distillation) stage. The system will be implemented in a modular, extensible manner, similar in spirit to recent open-source foundation model pipelines[19] but tailored to *C. elegans* imaging needs. Here we detail the key components and strategies:

- **Backbone Architecture:** At the heart of the model will be a **single backbone network** that processes input images (or video frames) and produces a rich

multi-scale feature representation. We will evaluate modern backbone designs that balance power and efficiency, such as a **Vision Transformer (ViT)** or **Swin-Transformer** backbone (which has shown excellent performance across vision tasks[28]), versus a **CNN-based** backbone (e.g. a ResNet or EfficientNet variant). The backbone will be **shared across all tasks**, embodying our hypothesis that common visual features (edges, textures, shapes, motion cues) can be learned once and reused. To handle multi-scale features (important for detecting both whole worms and fine details like neural structures), we may incorporate a Feature Pyramid Network (FPN) or UNet-style skip connections as part of the backbone output. The backbone will be designed to accept images from different modalities (brightfield, fluorescence) potentially by having separate input normalization or small modality-specific adaptation layers, ensuring compatibility with multi-modal data.

- **Multi-Task Heads:** On top of the backbone, we will attach a suite of lightweight **task-specific heads**, each tuned to the requirements of one analysis task. These heads will be trained jointly with the backbone, taking features from the backbone (at one or multiple pyramid levels) and producing task outputs. The planned heads include:

- **Semantic & Instance Segmentation Head:** We will implement a segmentation head that outputs pixel-wise labels for each worm and possibly other structures. To accommodate both semantic segmentation (classifying each pixel as "worm" vs background) and instance segmentation (distinguishing individual worms), we will explore a **panoptic segmentation** approach or a hybrid of **Mask R-CNN/Mask2Former** style architecture. For instance, the head might first predict object regions (e.g. bounding boxes or centroids for worms) and then generate a mask for each, or use a transformer-based segmentation that naturally handles overlapping instances[29]. This head will be supervised with ground-truth masks of worms (and could also output skeletons or keypoints along the worm if needed for posture analysis).
- **Object Detection Head:** In parallel to segmentation, an object detection head (e.g. anchor-free detector like FCOS or a YOLO-inspired head) will predict bounding boxes or centroids for worms (and possibly other relevant objects, such as eggs or specific organs if annotated). This provides coarse localization and counting. We anticipate that detection and segmentation tasks will reinforce each other – detection ensuring no worm is missed, segmentation refining precise outlines[15]. The detection head will output coordinates and confidence for each worm instance per frame.
- **Object Tracking Module:** Tracking will be achieved by a combination of model design and algorithmic association. We plan to incorporate a **tracking head** that produces an identity embedding or motion prediction for each

detected worm, enabling frame-to-frame association. One approach is to output, for each detection, a vector embedding (in a low-dimensional space) such that the same worm produces similar embeddings across time; these can be matched to maintain identities (similar to ReID in multi-object tracking). Another complementary approach is a small recurrent or transformer module that, given the backbone features at time *t* and *t+1*, predicts **offset vectors** or **optical flow** for worm positions. By integrating tracking into the network, rather than treating it as a separate post-processing step, the model can learn temporal consistency directly (e.g. smoothing segmentations over time). The tracking module will be trained with ground truth trajectories (IDs for worms across frames), possibly using metrics like the MOT (Multiple Object Tracking) accuracy or an association loss.

- **Super-Resolution Reconstruction Head:** For enhancing image resolution, we will include a **super-resolution (SR) head** that takes lower-resolution input frames (or deliberately down-sampled images) processed by the backbone and outputs a higher-resolution image (e.g. 2× or 4× upscaled)[30]. This head will likely use a decoder module that upsamples the backbone feature maps to the desired resolution. We may leverage architectures from super-resolution research (like ESPCN or RCAN) but crucially, the heavy lifting of feature extraction is done by the shared backbone. The SR head will be trained with paired low-res/high-res images of worms, optimizing for pixel-wise similarity (L1/L2 loss) and perceptual quality (e.g. using a feature-based loss to ensure realistic textures).
- **Denoising Head:** Similarly, a **denoising head** will be tasked with removing noise from images. We anticipate two modes: (a) **Brightfield denoising**, removing illumination artifacts or sensor noise from standard microscopy images, and (b) **Fluorescence denoising**, which is crucial for low-light neuronal imaging, etc. The denoising head might share architecture with the SR head (both are image-to-image mappings) – potentially we implement a single "restoration head" that can be toggled or conditioned for denoising vs. super-resolution. Training will involve either synthetic noise added to clean images (with the clean image as target) or real noisy-clean image pairs if available. Loss functions will include pixel reconstruction losses and possibly a noise-to-clean mapping loss (e.g. using techniques from Denoising Autoencoders or Noise2Void if unsupervised data is used). By training the model to internally represent a clean image, we expect the features used by other heads (segmentation, detection) will also become noise-invariant, improving their robustness.
- **Arbitrary Point Tracking (Feature Tracking) Head:** A unique capability we propose is an **arbitrary point tracking** function. This means the user (or an automated process) can specify any point or coordinate on the worm (or in the image) – for example, the position of a particular cell or a point on the

worm's body – and the model will track that point across time. To implement this, we will introduce a specialized head that can attend to a given point's feature and predict its new location in the next frame. One design is a **Siamese tracker**: the backbone features from frame *t* at the specified point (perhaps encoded as a small patch or a keypoint heatmap) are compared to features in frame *t+1* to find the best matching location (using correlation or attention mechanisms). We can integrate this by feeding the model a "point-of-interest" mask or coordinates as an input alongside the image, and have the head output a heatmap of the point's location in the next frame. Training data for this can be derived from known keypoint correspondences (e.g. tracking the head or tail of the worm, or fluorescently labeled neurons) – essentially any identifiable landmark can serve as training targets. This head would be especially useful for applications like tracking a specific neuron's position during movement, or following any user-tagged feature through time without needing a full segmentation.

- **Training Strategy:** The model will be trained in a **multi-task learning** paradigm, where all the above heads are optimized simultaneously. We will formulate a **composite loss function** that is the weighted sum of losses from each task head: e.g. $\mathcal{L} = \lambda_{seg}\mathcal{L}*{seg} + *lambda*}\mathcal{L*{det} + *lambda*}\mathcal{L*{track} + *lambda*}\mathcal{L*{sr} + *lambda*}\mathcal{L*{denoise} + *lambda*$. Choosing the weights λ is non-trivial; we will likely start with equal weighting after normalizing each loss (such that each loss term is of order 1 at the start, following methods used in multi-task models like U}\mathcal{L}_{point**Net** with multiple heads[31][32]). We will also explore **dynamic loss weighting** schemes, for example adjusting weights based on task difficulty or using uncertainty-based weighting (where the model learns per-task weight parameters). Our optimization will use backpropagation on the unified network, possibly with **task-specific batch sampling** (ensuring that each mini-batch contains data for a mix of tasks or cycling between tasks if memory doesn't allow all at once). We may employ techniques to mitigate **negative transfer**: one idea is **gradients modulation or decoupling** – e.g. using separate optimizer steps for different task groups, or approaches like **gradient surgery** to remove conflicting gradients between tasks. Another is the use of **auxiliary single-task networks as teachers** during training (related to "online distillation" for multi-task learning[33][34]). In fact, if we find certain tasks lagging, we could train a single-task expert for that task and use it to guide the multi-task model via a distillation loss during training[35][36], to improve that task's performance without sacrificing others. Overall, the training will likely proceed in **stages**: an initial phase focusing on core vision tasks (segmentation/detection/tracking), and later phases incorporating the image reconstruction tasks (SR/denoise) which might require longer training or careful scheduling (since they are regression

tasks with potentially high-dimensional outputs). We will also consider a **two-stage training** where first the backbone is pre-trained (see next bullet) and initial heads are trained, then additional heads (like SR) are added and fine-tuned.

- **Pre-training and Initialization:** To give our model a strong starting point (especially important given the relatively limited size of labeled worm datasets), we will leverage pre-training. There are a few strategies:

- **Supervised pre-training on generic vision data:** We could initialize the backbone with weights from ImageNet classification or, more pertinently, from a **general segmentation model** (e.g. using SAM's or BiomedParse's encoder if available). For example, SAM's ViT-H encoder trained on billions of masks encodes a wealth of segmentation knowledge[8]. Using a smaller variant of SAM's architecture pre-trained on COCO or other images might accelerate convergence and improve segmentation of worms. We must be cautious to avoid any license or model size issues (SAM is huge), so we might use a smaller ViT (like ViT-B or Swin-B) pre-trained on large datasets as our backbone starting point.
- **Self-supervised or weakly-supervised pre-training on worm data:** We have the option to gather a large number of unlabeled *C. elegans* images and videos (which are easier to obtain than labeled data) and perform self-supervised learning (e.g. contrastive learning, masked image modeling) to pre-train the backbone. This could imbue the model with an understanding of worm-specific structures without manual labels. Additionally, we could exploit **synthetic data generation** for worms (as WormSwin did, generating composite images to simulate overlapping worms[37]) to create a massive pre-training set. Weak labels – such as "does an image contain worms or not" or "rough region proposals by thresholding" – could also guide a pre-training stage.
- **Task-specific pre-training:** Another route is to sequentially pre-train on easier tasks. For instance, begin with basic segmentation on a small dataset to tune the backbone to worm shapes, then introduce detection/tracking on video data, and finally the more complex tasks (SR, etc.). This staged approach can stabilize training, ensuring the backbone isn't overwhelmed by too many tasks at once initially.

We will evaluate these approaches empirically, aiming for one that maximizes backbone general features without overfitting to any single task or domain.

- **Knowledge Distillation Pipeline:** Once the **high-capacity model (teacher)** is trained and achieves strong multi-task performance, we will initiate a model compression phase to produce a **deployable student model**. The teacher model might have a large backbone (for instance, a transformer with tens of millions of parameters), whereas for the student we will target a

much smaller backbone architecture (e.g. a MobileNetV3, EfficientNet-B0, or a Tiny-ViT with drastically fewer parameters). The student will retain the same set of task heads (or slightly simplified versions) so that it outputs the same types of predictions as the teacher. Our distillation strategy will include:

- **Feature distillation:** We will encourage the student's intermediate feature maps to match the teacher's. For each relevant stage in the backbone (and possibly for each task head's internal features), we will use losses like mean squared error (MSE) or a **perceptual loss** to align the student's representation with the teacher's[30][38]. This helps the student capture structural and semantic information that the teacher has learned[30]. For example, we might project the teacher's and student's feature maps to a common dimensionality and minimize the difference between them (this can be done at multiple scales).
- **Output distillation:** In addition to features, the final outputs of the student will be trained to mimic the teacher's outputs for each task. For classification-type outputs (detection confidences, segmentation logits), we can use the classic distillation loss (softened cross-entropy aligning the student's probability distribution to the teacher's). For regression outputs (bounding box coordinates, point tracking, pixel regression in SR/denoise), we will minimize L1/L2 loss between student and teacher outputs. The ground-truth labels will also be used, but the teacher's outputs serve as an additional learning signal – effectively, the student is taught to produce what the teacher would produce, which smooths out noise and provides guidance on ambiguous cases.
- **Decoupled encoder/decoder training:** Inspired by recent work on accelerating SAM for medical imaging[39][40], we may adopt a two-phase distillation: first train the student backbone (encoder) to emulate the teacher's backbone (possibly freezing teacher heads and just matching features), then train the task heads with the teacher's outputs fixed. This decoupling can simplify optimization and has been shown to improve student performance when dealing with very different backbone sizes[40].
- **Online vs. offline distillation:** We will compare offline distillation (train teacher fully, then train student on recorded teacher outputs) versus online (train student alongside teacher, or using a pre-trained teacher but generating outputs on the fly). Online distillation in multi-task settings, sometimes called "rocket launching," has had success in guiding lightweight models by a heavy model in real-time[41][42]. In our case, offline might be simpler (since we'll have a fixed teacher), but online could be used if we decide to train a student from scratch with occasional teacher guidance to reduce training time.
- **Optimization and Metrics:** The distillation training will use similar losses as the original, but with an emphasis on matching the teacher. We'll carefully

monitor the student's performance on validation sets for each task, aiming to minimize the drop from teacher to student. The expectation, based on prior art, is that the student can achieve **comparable or even slightly superior performance** on some tasks due to the regularizing effect of distillation[27]. For example, KD-SAM (knowledge-distilled SAM) was able to achieve comparable segmentation accuracy to SAM with far fewer parameters[27]. We hope to replicate such success across multiple tasks.

- **Computational Considerations:** Throughout development, we will pay attention to model efficiency. The final student model will be benchmarked for **inference speed** (frames per second processed) and memory usage. Design choices like using depthwise separable convolutions, low-bit quantization (post-training quantization or quantization-aware training), and model pruning will be considered if needed to reach real-time performance. One advantage of our approach is that a single forward pass through the backbone can produce outputs for all tasks, which is more efficient than running separate models for each task sequentially. For instance, where a traditional pipeline might run a segmentation CNN, then a tracking algorithm, then a denoising filter separately, our model would do one backbone computation and feed all heads in parallel – this shared computation should save time overall[43]. We will validate that the multi-head inference does not become a bottleneck (e.g. we may parallelize head computations on different GPU streams or optimize the most demanding head).

- **Validation During Development:** We will adopt rigorous validation at each stage. For example, after initial multi-task training, we'll compare the multi-task model against single-task models on held-out data to check for any performance degradation. We will also perform **stress tests**: e.g. extremely low light images to test denoising + segmentation, or heavily populated worm videos to test segmentation + tracking coherence. Any identified weakness will inform adjustments (like adding a specific loss term or data augmentation to handle that scenario).

By combining a carefully designed architecture with multi-task learning and a distillation pipeline, this methodology aims to produce both a **powerful "teacher" model** that proves what is achievable (for benchmark and hypothesis-testing purposes) and a **practical "student" model** that can be widely used in biological labs. The novelty lies in unifying such a wide range of tasks in one model for a specific organism, leveraging ideas from foundation models (joint training on diverse tasks) but also ensuring **pragmatic deployment** via model compression – an aspect often overlooked in foundation model research[21]. This framework will be one of the first to integrate everything from low-level image enhancement to high-level object tracking for *C. elegans*, pushing the frontier of both worm research and multi-task deep learning.

Data Resources and Integration

A critical aspect of this proposal is the **collection and integration of diverse datasets** covering the spectrum of *C. elegans* imaging tasks and modalities. We will leverage both **publicly available datasets** and **proprietary data** from our laboratory, ensuring comprehensive coverage:

- **Existing Public Datasets:** Several open datasets and resources will be incorporated:
- *Worm segmentation datasets:* We will use the Broad Bioimage Benchmark Collection (BBBC) worm datasets, such as **BBBC010**, which contains images of *C. elegans* (potentially from high-throughput screens) with ground truth labels[12]. Additionally, the **WormPose/NemaDataset** from Javer *et al.* (2018) provides thousands of annotated worm images and videos (available on Zenodo) focusing on worm posture and locomotion features[7]. The **SegWorm** dataset and **Tierpsy Tracker** outputs might also provide segmentation masks and skeletons for worms in various scenarios. We will aggregate these to form a robust training set for segmentation and detection.
- *Worm tracking and behavior datasets:* We will incorporate datasets from studies like **Open Worm Movement Database** or those used in multi-worm tracking challenges. The recent PLOS Comp Bio study by Liu *et al.* (2025) provides a dataset (via their GitHub[44]) of videos with multiple worms along with detection/tracking ground truth and extracted behavior metrics. We will use those videos and annotations (e.g. worm bounding boxes and trajectories) to train and validate the tracking components. Another source is the **Multi-Worm Tracker (MWT) dataset** (from Swierczek et al. 2011), which, while based on older methods, contains video recordings of worms that we can re-annotate or use for unsupervised training.
- *Fluorescence microscopy and neuron imaging:* For modeling fluorescence data, we will utilize any available datasets of *C. elegans* neural activity imaging. For example, the **NeuroPAL** dataset provides images of worms with many neurons uniquely colored (though static) – these can help train segmentation of neural regions. More directly, we have the data from Kato *et al.* (2015) or other whole-brain imaging experiments (if accessible) where worms expressing GCaMP (a calcium indicator) were recorded; these often have a few neurons fluorescing and can be used to train the model on segmenting or tracking neural signals amid body movement. If published data are insufficient, see below for our own contributions.
- *Super-resolution / multi-resolution data:* There are fewer public worm datasets explicitly for super-resolution. However, we can exploit any dataset that has the same scene imaged at different magnifications or resolutions. For example, the **HTDA (High-Throughput Drug Assay) dataset** might include both wide-field (low-res) and zoomed-in (high-res) images for certain

worms; if not, we can simulate this by downsampling high-resolution images to create training pairs. The BBBC or other screening datasets often contain low-resolution but high-count images, which can be paired with a small number of high-res images of the same worm strains for SR training. We will also use natural image augmentation (e.g. blurring to simulate low-res) for generating synthetic SR training pairs.

- *Denoising data:* Public datasets for denoising might include sequences where an image is acquired multiple times (averaging can produce a ground truth). If not directly available for worms, we might use general microscopy denoising benchmarks (like fluorescence microscopy denoising datasets from Cell Tracking Challenge or others) to pre-train. For brightfield, image time-series with static scenes can be used (treating the averaged image as clean). We will search for any microPublication or supplemental dataset focusing on worm image denoising; in absence, rely on synthetic noise addition on real images for training.

Importantly, each public dataset comes with its own format and biases (different labs, strain variants, camera settings). We will unify these in a common format (likely converting all images to a standard pixel scale and bit-depth) and ensure the **metadata** (like frame rate, pixel size) are recorded for use in analysis (e.g. converting pixel speeds to real units).

- **Proprietary and New Data Collection:** To complement public data and fill any gaps, we will use and generate data from our laboratory:
- *High-Throughput Behavior Videos:* We have access to a multi-worm tracking rig that records dozens of worms simultaneously in large agar plates (brightfield illumination). We will produce a set of videos with varying worm densities, including challenging scenarios (crowded conditions, worms overlapping, different life stages from L1 larvae to adults to test generality). These videos will be annotated by our team: we will use semi-automatic annotation where possible (e.g. run SAM to get initial masks[8], then have human annotators correct them, thereby rapidly generating segmentation labels). For tracking ground truth, we will manually assign IDs to worms in a subset of frames and propagate these via an existing tracker, then correct identity switches manually. This will yield a labeled dataset for multi-worm tracking in conditions resembling real lab experiments.
- *Fluorescence Neural Imaging Coupled with Behavior:* In collaboration with our neuroscience colleagues, we will record *C. elegans* with a calcium indicator (GCaMP) in specific neurons (e.g. the RIA interneuron or a motor neuron) while simultaneously capturing the worm's posture. For instance, worms may be constrained to a microfluidic device or a small chamber where they can move their head/tail while neural activity is imaged. We will collect video data in two channels (or sequentially): one brightfield (for the worm's body) and one fluorescence (for neurons). These videos will allow us

to create ground truth for the **point tracking** task (tracking a specific neuron's location) and test cross-modal performance. We will annotate neuron positions over time (possibly by thresholding the fluorescence and linking spots) to provide training targets. Additionally, we will obtain expert segmentation of the worm's body and the neuron region in some frames (similar to TWARDIS which segmented neural compartments[20]).

- *Super-Resolution Image Pairs:* Using our microscope, we will capture the same field of worms in two settings: a "low resolution" setting (e.g. using a 4x or 10x objective, or a lower camera resolution) and a "high resolution" setting (e.g. 20x or 40x objective, or simply digital zoom with finer sensor resolution). By doing this for many fields and timepoints, we can build a paired dataset: each low-res image has a corresponding high-res version as ground truth. We will ensure alignment (using fiducial markers or image registration) so that the model can learn to map one to the other. This will directly supervise the super-resolution head. If needed, we will also include cases of motion blur vs. sharp images to account for temporal resolution enhancements.
- *Noisy vs. Clean Image Data:* For brightfield denoising, we can take images under lower illumination or faster shutter (introducing noise) and compare with longer exposure or averaged frames. For fluorescence, we will record some sequences with low laser power (noisy) and then with high laser power or averaged multiple runs (clean), to create training pairs. We might also use denoising algorithms as a pseudo-ground-truth generator: e.g. run a strong non-local means or BM3D on a noisy image to get a "cleaned" reference. These strategies will enlarge our denoising training set.

All new data collected will be carefully annotated. We will employ an annotation strategy that maximizes consistency across annotators and tasks: - For segmentation, we will develop a standard protocol (outlining whether certain touching worms are separated or counted as one, how to mark partially out-of-frame worms, etc.). We may use **annotation tools like CVAT or Labellng** and train our team (or crowdsource if feasible) to label worm outlines. As mentioned, we'll accelerate this with AI assistance (SAM, etc.), which can yield masks that we then correct[45]. - For tracking ground truth, besides manual labeling, we can use an approach of synthetic trajectory generation for validation: e.g. overlaying a moving worm cutout on a background to test tracking, but primary focus will be real video annotations. - Each image or video will be tagged with domain information (modality, imaging conditions) so that we can later analyze performance per domain and possibly incorporate domain-specific normalization (like different intensity scaling for fluorescence images). - We will also split data into **training, validation, and test sets** from the start, using hold-out labs or conditions for testing cross-domain generalization. For example, videos from one microscope will be train, and a video from a completely different microscope (unseen during training) will serve as a test to evaluate generalization.

- **Data Integration and Management:** We will create a unified **data repository** (potentially on a platform like DataDryad or our institution's server) that consolidates all these resources. Each data instance will be formatted into a consistent structure so that the model can ingest it. For multi-task training, we'll need to handle heterogeneous data: some images have segmentation labels but not super-resolution pairs, some videos have tracking but not fluorescence, etc. We plan to use a multi-task data loader that can sample from different pools. For example:
- When sampling a batch for segmentation/detection, use an image (or video frame) that has segmentation masks and possibly detection annotations.
- For tracking, sample sequences of consecutive frames with worm ID annotations.
- For super-resolution, sample a low-res image and its high-res counterpart.
- For denoising, sample a noisy image and its clean target.

We might use **curriculum learning** or alternating cycles: e.g. one epoch focusing on segmentation/detection, next on tracking, etc., or mix them every batch (adjusting learning rates as needed). We will ensure that over an epoch, each task's entire dataset has been seen proportionally so no task is starved of learning. Loss normalization (as discussed) will help integrate these different tasks.

- **Data Volume and Diversity:** We anticipate the aggregated dataset will be substantial: potentially on the order of tens of thousands of images (segmentation/detection) and dozens of hours of video (for tracking). The textual descriptions are not a focus here (unlike some foundation models that also incorporate text[17]), as we are centering on vision tasks; however, we will document the context of each dataset to interpret results (for instance, noting if certain strains or behaviors are challenging). Diversity is key – *C. elegans* can vary in appearance (e.g. brightfield vs darkfield imaging, presence of bacteria on plates causing noise, etc.), so we include such variations. If some modalities are underrepresented in public data, we will augment them via synthetic means (e.g. adding artificial occlusions or background clutter to images to simulate difficult conditions, or using GANs to simulate fluorescence patterns).

By combining these resources, we aim to provide the model with a **rich and varied training experience**, covering the full range of tasks and conditions it will encounter. This comprehensive data integration is a cornerstone of the project – it will also serve as a valuable **benchmark suite** for the community. Indeed, beyond developing the model, one outcome will be the curated multi-task dataset for *C. elegans*, which we plan to release openly alongside publication, contributing to standards in the field.

Systematic Experimental Plan

A rigorous experimental plan will be executed to evaluate the proposed model against our hypotheses. We outline a series of experiments and analyses, each aimed at validating specific aspects of the model's performance, adaptability, and efficiency:

1. Multi-Task Performance Benchmarking

Objective: Test whether the multi-task foundation model achieves comparable or better performance on each individual task relative to dedicated single-task models (addressing Hypothesis H1).

Design: For each task (segmentation, instance segmentation, detection, tracking, super-resolution, denoising, point tracking), we will establish baseline models: - For segmentation: use a state-of-the-art worm segmentation model such as WormSwin (ViT-based)[29] or a classic U-Net trained specifically on our worm segmentation data. - For detection: a YOLOv8 model or Faster R-CNN trained purely for worm detection (some of these we have from prior work). - For tracking: the YOLO+ByteTrack pipeline from Liu *et al.* (2025) as a baseline[11], and possibly the multi-animal tracker (MAT) or other algorithmic trackers as another reference. - For super-resolution: a standard super-resolution network (like EDSR or RCAN) trained on the worm SR pairs. - For denoising: a UNet or DnCNN trained for denoising. - For point tracking: if no existing direct model, we use an optical flow method (like DeepFlow or a Lucas-Kanade tracker on enhanced images) as a baseline for tracking arbitrary points.

We will evaluate the **teacher model** (full-size backbone) on held-out test sets for each task: - Segmentation: measure IoU (intersection-over-union) or Dice coefficient of worm masks, and count the number of segmentation errors (e.g. worms missed or merged). - Instance segmentation: measure average precision at IoU 0.5 (AP@0.5) and AP across IoU thresholds, as in COCO evaluation, to gauge how well individual worms are separated[46]. - Detection: measure precision, recall, mAP (mean average precision) for worm localization. The PLOS study reported 99.6% mAP50 for their detector[11]; we will see where ours stands in comparison. - Tracking: use standard multi-object tracking metrics – e.g. MOTA (Multiple Object Tracking Accuracy), MOTP (Precision), ID switch counts, track fragmentation counts – on benchmark videos. If possible, we'll include a scenario like the one in Liu *et al.* (2025) and directly compare metrics (they achieved high continuity and robustness, nearly no ID switches at moderate densities)[11]. - Super-resolution: compute PSNR (peak signal-to-noise ratio) and SSIM (structural similarity) between the model's super-res output and ground truth high-res images, as well as perceptual quality metrics (maybe using a pre-trained network to assess feature similarity). We will also visually inspect if fine details (like thin worm structures or textures) are correctly reconstructed. - Denoising: measure improvement in PSNR/SSIM from

input noisy image to output, and ensure that biological details (like small worm features or fluorescence spots) are preserved. We might also feed denoised images to a segmentation algorithm to see if segmentation accuracy improves, demonstrating practical value. - Point tracking: measure the average tracking error (distance between predicted and true point positions over time) and success rate (percentage of time a point is within an acceptable radius of the true location). Compare against optical flow or manual tracking.

Success Criterion: The multi-task model should perform at least on par with single-task baselines on majority of metrics. We anticipate especially strong performance in integrated tasks: for example, the multi-task model might surpass a standalone segmentation model on images with heavy noise or occlusion, because it learned from tracking and denoising tasks how to handle those[10]. If any task is noticeably worse, we will investigate and perhaps adjust training (e.g. increase that task's loss weight or improve its head design) and rerun. This experiment directly tests if our unified approach introduces any **regression in capability** relative to specialized approaches.

2. Negative Transfer vs. Synergy Analysis

Objective: Determine whether multi-task learning led to any negative transfer (one task's performance degraded compared to single-task training) or positive synergy (multi-task model outperforms single-task on certain tasks or scenarios). This further probes Hypothesis H1 and H4.

Design: Using results from Experiment 1, we will analyze on a per-task basis: - Compare the multi-task model's accuracy to the single-task model's accuracy for each task on identical test data. Quantify the percentage difference. - Conduct **statistical tests** (if applicable) to see if differences are significant (e.g. a paired t-test on frame-level segmentation accuracies between models). - Identify **scenarios where differences are pronounced**: e.g. does the multi-task model especially outperform on noisy images (suggesting denoising task helped)? Does it perhaps underperform on extremely simple images (maybe it overfits to complexity)? We will stratify results by data attributes: low vs. high noise, single worm vs. many worms, etc.

Additionally, we will perform **ablation studies**: - Train ablated versions of the model with one task left out at a time. For example, train a model without the denoising head (all other tasks still present). Then evaluate its performance on segmentation, detection, etc., especially on noisy data, to see if leaving out denoising hurt those tasks. Likewise, train without tracking and see if segmentation/detection suffer in videos (which would indicate the tracking task was helping learn temporal consistency). - Conversely, test a model without a certain task on that task's own metrics as a sanity check (e.g. if we train without super-resolution, obviously it cannot do SR; but perhaps the segmentation quality might be slightly higher

because capacity wasn't used for SR – this would indicate some resource contention).

From these ablations, we can infer **task interdependencies**: - If removing task A worsens performance on task B, then task A was providing beneficial regularization or features for B (positive synergy). - If removing task A actually improves B, then maybe task A was interfering (negative transfer). - We expect mostly synergy or neutral interactions, but this will confirm and highlight any necessary mitigation (like better loss balancing if a conflict is found).

We will document, for example, "Including the **denoising task improved segmentation IoU by X% on noisy images compared to a model trained without denoising, supporting that multi-task learning provides robustness to noise" or "Including tracking slightly decreased segmentation accuracy on static images by Y%, indicating a slight negative transfer which might be due to the model focusing on motion features – a trade-off we address with… (if needed)."

Success Criterion: Ideally, we find that no task's inclusion drastically harms another's performance, and in several cases there are clear benefits (e.g. joint training with multiple tasks increases generalization as measured in Exp 3 below). If any major negative transfer is detected, we will adapt (for instance, if super-resolution training is hurting other tasks, we might compartmentalize its influence by training SR head in later epochs or reducing its weight). This analysis guides fine-tuning of the training strategy to maximize synergy.

3. Cross-Domain and Generalization Evaluation

Objective: Assess the model's ability to **generalize across domains** and modalities that were not explicitly matched in training, addressing Hypothesis H2.

Design: We will prepare several domain-shift scenarios: - **Cross-Lab Generalization:** Train the model on images from our lab and some public data, then test on an entirely independent dataset from another lab that the model never saw. For example, we might exclude the BBBC010 dataset during training and then test on it as an external set, or vice versa. Metrics: measure segmentation AP or tracking accuracy on this external data, and compare to (a) our model's performance on similar domain data it was trained on, and (b) any published results by others on that dataset (if available, e.g. WormSwin reports an AP on BBBC010[12]). - **Modality Generalization:** Evaluate performance when the imaging modality differs. For instance, use our model trained on mixed data to segment fluorescent images (where worms appear as bright objects on dark background) even if only a smaller portion of training data was fluorescent. Compare against a baseline that was trained purely on fluorescence images (if we train one). We may intentionally hold out one modality during training to stress-test; e.g. train on brightfield only, test on fluorescence, to see how much of a drop occurs when domain is changed. Then compare to multi-modal trained model's performance on the same test to quantify

improvement from multi-modal training. - **Robustness to Experimental Variations:** This includes generalization to different worm strains (wild-type vs. mutants that might have different morphology or movement patterns), different imaging conditions (lighting intensity, presence of debris or other organisms in the field, etc.). We will simulate or use real data for these: e.g. test if the model can detect worms in a video where the background has food or bubbles which could confuse a less general model. - **Low-Data Regime Test:** Another dimension: if possible, fine-tune or train a small model on a very limited dataset from a new domain and see how pre-trained foundation model helps. For example, if someone images worms in a novel assay (say microfluidic chips), can our foundation model (perhaps fine-tuned lightly) adapt quicker than training from scratch? This would demonstrate the value of a broad backbone. Practically, we can take a small sample from a new domain, fine-tune our model vs. fine-tune a baseline model, and measure performance to show improved data efficiency.

Success Criterion: The multi-task foundation model should show **strong cross-domain performance**, e.g. maintaining high accuracy on external or varied data without needing retraining. If, for instance, a single-modality model fails (IoU drops drastically on a new modality) but our multi-modal trained model retains much higher IoU, that validates H2. We will quantify improvement. For example, BiomedParse showed improved accuracy especially for irregular objects and could segment across nine modalities with one model[47] – we expect similarly that our model's error on a new domain is significantly lower than a collection of single-task models not trained on that domain. If weaknesses are found (say the model struggles on a particular type of fluorescence artifact), that will guide us to incorporate a bit of that domain in training or adjust input normalization to account for it. But overall, our aim is that **no single domain shift catastrophically breaks the model**, demonstrating it as a true foundation model for worms.

4. Efficiency and Deployment Evaluation

Objective: Evaluate the lightweight student model's performance vs. the teacher and measure deployment-relevant metrics (speed, memory), addressing Hypothesis H3.

Design: After training and distillation: - **Performance Retention:** Measure all the task metrics for the **student model** on the test sets, side by side with the teacher model's metrics. We expect slight drops in some metrics, but we'll quantify exactly. For example, if teacher segmentation mIoU = 0.90 and student = 0.88, that's ~2% drop, which might be acceptable. We'll see if any particular task suffered more; for instance, did the student struggle more in super-resolution detail or tracking consistency? Identify if any distillation loss weighting needs adjustment – we might iterate on distillation if one task lost too much accuracy (e.g. give it a higher weight in knowledge transfer). - **Speed Benchmarking:** We will benchmark inference speed of both teacher and student models: - On a high-end GPU (for reference, say

an NVIDIA A100 or V100), measure FPS (frames per second) the model processes for a typical video (including all heads). The teacher might achieve, hypothetically, 5 FPS whereas the student might target >30 FPS. We'll also measure on a modest GPU (like a common desktop GPU, e.g. GTX 1650 or similar) and on CPU if feasible. - If real-time processing is needed (e.g. 10 FPS for certain experiments), check if the student meets it. The PLOS 2025 model hit 153 FPS on a strong GPU for detection/tracking[11]; our student might not be that high given additional tasks, but we aim for real-time on at least GPU. - We will document memory usage (VRAM) and model file size as well. Perhaps our student ends up at e.g. 50 MB, versus teacher 500 MB – which is a significant reduction. - **Ablation on Student Complexity:** We might train multiple student variants (e.g. one even smaller than target, one slightly larger) to see the trade-off curve. This helps justify our choice of student architecture: if a much smaller student results in unacceptable accuracy loss, we'll know the threshold. - **Edge Device Test:** If resources allow, deploy the student model on an edge device (such as an NVIDIA Jetson Nano or a laptop without GPU) and attempt to analyze a sample video. Record the processing rate and any issues. This will demonstrate practical deployability – for high-throughput experiments, being able to run on an instrument computer is important.

Success Criterion: We consider the distillation successful if the student model retains **the vast majority of the teacher's performance** (for example, >90% of the teacher's accuracy metrics on all tasks) while being significantly faster and smaller. Achieving **real-time or near-real-time speeds** on standard hardware will be a key benchmark. If the student falls short in any area, we will refine the distillation (for example, adding a perceptual loss improved KD-SAM results[48][38], so we might include similar losses if our initial student had noticeable qualitative issues in segmentation boundaries, etc.). Ultimately, this experiment will validate that our approach yields a **practically usable model** without sacrificing the quality proven by the teacher model.

5. End-to-End System Validation in Use-Case Scenarios

Objective: Demonstrate the model's integrated capabilities in realistic application scenarios, tying together multiple tasks and showing qualitative and quantitative improvements over existing workflows (addresses Hypothesis H5).

Design: We will conduct a set of case studies using the **entire pipeline** of our model on real experimental data: - **High-Throughput Drug Screening Assay:** Take a dataset from a drug screening experiment (for instance, worms treated with various compounds, imaged in multi-well plates over time). We will run our model to automatically segment all worms in each well, count them, track their motion, and extract behavioral metrics (speed, bending, etc.)[11][49]. We will compare this to the traditional analysis: e.g. using Tierpsy Tracker or manual counting. Metrics: time taken to process a plate, number of worms correctly identified (sensitivity to find all worms vs false positives), and the consistency of measurements. We expect our

model to significantly improve throughput (maybe reducing what was hours of analysis to minutes) and handle tricky cases (overlapping worms) without excluding data[10]. If possible, we'll quantify improvement in statistical power – for example, if a drug slows worm movement, the model's tracking data should detect that with less variance than manual or simpler methods, meaning fewer replicates needed. - **Transgenic Behavior Phenotyping:** Use data from a strain of worms with a neural or muscular mutation that causes subtle behavioral changes. We will apply our model to measure features of motion (e.g. amplitude of body bends, reversal frequency) and see if we can automatically distinguish mutant vs wild-type phenotypes. Such analysis might normally require careful parameter tuning per strain; we will show our foundation model can robustly extract features across strains. If there is a known phenotype (say mutants are slower or have different posture distributions), we'll verify the model captures that difference. This validates that the model is reliable as a **standardized tool** for phenotyping, which is often needed in genetics studies. - **Neural Activity Coupling Example:** Take the fluorescence+behavior dataset (e.g. imaging neuron RIA activity while worm moves) and use our model's segmentation + point tracking outputs. Specifically, segment the worm's body in the fluorescence video, track the worm's head position (or the neuron itself) and extract fluorescence intensity over time in that neuron region. Compare our automatic approach to the conventional approach (where one might manually draw a region-of-interest (ROI) or use simple thresholding on each frame). We expect our model to more accurately isolate the neuron signal because it can segment the worm's body and discount movement artifacts, yielding a cleaner calcium signal trace[20]. We'll demonstrate how the model's ability to get "biologically accurate, absolute head positions" alongside neural signals allows analysis of neuron-behavior correlation that was previously hard to do quantitatively[20]. For instance, we can compute the correlation between neuron activity and specific behaviors (like turns or accelerations) using our tracked motion data, showing new insights. - **Image Enhancement for Microscope Imaging:** To illustrate the benefit of the super-resolution and denoising tasks, we will simulate a resource-saving scenario: capture a worm video at lower magnification and higher noise (which allows larger field of view and faster imaging) and use our model to enhance it. Then compare the outcome to a video captured at high magnification/low noise directly. We expect the model's SR and denoise heads to reconstruct much of the detail, enabling, for example, posture analysis on the enhanced video that is as good as on the high-quality video. This shows the potential for using cheaper imaging setups with AI enhancement to still get high-quality data, which could be very useful for scaling experiments.

For each scenario, we will document: - Qualitative results (images, example frames with our model's segmentation or tracking overlays, before-and-after denoising comparisons, etc.). - Quantitative results (e.g. number of worms tracked through entire experiment with no loss, behavior metrics significance, neuron signal noise-

to-signal improvement). - We will also gather user feedback if possible: e.g. have a worm researcher use our software vs. existing, to report on usability improvements.

Success Criterion: The integrated model should prove its value by **simplifying analyses and revealing data that was previously obscured**. For example, if our model can automatically handle overlapped worms, we should show that we get 100% usage of video frames, whereas previous methods might have dropped X% of frames or required manual cleanup[10]. In drug screening, success might be demonstrated by detecting a drug effect with our automated measurements that manual scoring missed, or doing it in a fraction of the time. Ultimately, these real-world tests will confirm that our approach meets the high standards of **Nature Methods**-level work: not just algorithmically novel, but a clear advance for experimental science.

Any shortcomings noted in these tests (e.g. the model might have minor failure modes in extremely unusual conditions) will be analyzed and reported transparently, along with suggestions for future improvement (like adding more training data or a specific module to handle that case). Since our aim is a **publication-quality study**, we will ensure that all claims are backed by thorough experiments, with significance tests or multiple replicates where appropriate to demonstrate robustness.

In summary, this experimental plan will validate the proposed model from all angles: per-task accuracy, multi-task synergy, generalization across conditions, computational efficiency, and practical utility. Through these experiments, we expect to establish that our *C. elegans* foundation model meets its design goals and offers a compelling advantage over the fragmented approaches currently in use.

Application Examples and Impact

A unified, deployable *C. elegans* foundation model opens up numerous high-impact applications in biology. Here we highlight several example use cases to illustrate the model's value, each aligning with pressing needs in the field:

- **High-Throughput Drug Screening:** *C. elegans* is widely used in phenotypic screens for drug discovery and toxicology, where researchers observe how thousands of worms respond to chemical compounds. Our model can dramatically enhance such screens by **automating worm detection, health assessment, and behavior tracking in multi-well plates**. For instance, in a typical antiparasitic drug screen, worms might need to be counted (live vs dead) and their movement quantified under each drug dose. Using our foundation model, an entire plate's images can be processed in seconds to segment every worm, even in wells with dense populations, with near-human accuracy[11]. The model's tracking head would follow individual worms over time, providing readouts of locomotor activity, while the denoising/SR heads ensure even suboptimal images are analyzed reliably.

This obviates labor-intensive manual scoring and yields **standardized metrics** across experiments[50]. As a concrete example, Liu *et al*. (2025) noted their deep learning tracker greatly improved throughput and enabled simultaneous extraction of multiple behavior parameters, facilitating drug screening studies[49]. Our model extends this by handling additional complexities (like worms overlapping or poor image quality) within one framework. The impact is a **significant increase in throughput and consistency** – screens can include more compounds or replicates since analysis is no longer a bottleneck, and subtle drug effects on behavior (e.g. slight reductions in speed or changes in locomotion patterns) will be quantifiable with high precision[11][49].

- **Transgenic Worm Phenotyping and Mutant Analysis:** Researchers often create transgenic worm strains (for example, knocking out a gene or expressing a fluorescent reporter) to study gene function. These strains might have subtle **behavioral or morphological phenotypes** that are hard to measure. Our model provides a comprehensive phenotyping tool: the **segmentation head** yields precise morphology (size, shape) of worms, allowing morphometric comparisons (e.g. slight body length differences or fat storage levels if imaged)[10]. The **tracking and pose analysis** (via segmentation+tracking) quantifies locomotor patterns – for instance, a neural mutant might have shorter or more erratic movements, which our model can detect by measuring path straightness, bending frequencies, etc. Moreover, because the model is multi-modal, it can analyze **fluorescent reporters** in the same individuals: e.g. a strain expressing GFP in neurons could be analyzed such that the model segments the worm and also reads the fluorescence intensity in those neurons (by integrating segmentation with intensity measurement). This unified analysis means we can correlate a mutant's behavior with reporter signals or subtle anatomical changes simultaneously. **Importantly, the foundation model's pretraining on diverse wild-type data can act as a reference**, helping to highlight deviations in a mutant. This reduces bias and makes phenotyping more sensitive and objective. In practice, our tool could become part of the standard workflow for characterizing new *C. elegans* strains – plugging in a video of mutant vs. wild-type worms and getting a rich report of differences in movement, growth, etc., all with rigorous statistical backing.

- **Neural Activity and Behavior Coupling:** One of the grand challenges in neuroscience is to link neural dynamics to behavior. In *C. elegans*, with only 302 neurons, researchers can simultaneously record neuronal calcium activity and the worm's movements. Our model is uniquely suited to this integrative task. Using the **point tracking head**, a researcher can mark a neuron of interest (observed via calcium fluorescence) and automatically track its position within the moving worm across frames. Simultaneously,

the **segmentation and tracking heads** delineate the worm's posture and motion. This yields, for the first time in an automated way, a synchronized readout of "what the neuron is doing" and "what the worm is doing". For example, our model would allow analysis like: *during an omega turn (a deep bend behavior automatically classified from posture metrics), what is the activity of the RIA interneuron?* TWARDIS demonstrated that with proper segmentation, one can avoid averaging signals over a moving ROI and instead get precise neural signals aligned with actual head movement[20]. Our model would extend that principle: every frame, it can output the worm's exact head angle and the neuron's fluorescence, enabling calculations of correlation or lead/lag relationships. This can lead to discoveries about how neural circuits drive behavior (e.g. identifying neurons that spike right before a reversal movement). Additionally, because our model can handle **semi-restrained or free-moving worms** (thanks to its robustness across modalities and time), experiments that were previously limited by analysis (like tracking neurons in freely crawling worms) become feasible. The expected impact is a deeper understanding of sensorimotor integration in *C. elegans*, achieved with a turnkey analysis pipeline rather than months of manual data curation.

- **Improved Microscopy and Experimental Design:** Beyond direct data analysis, our model could influence how experiments are conducted. Its **denoising and super-resolution capabilities** mean that researchers could opt for gentler imaging conditions (lower light, faster acquisition, lower magnification) and rely on AI to enhance the data. For example, in long-term imaging of developmental processes, phototoxicity and data volume are concerns. Using our model, one could capture sparse, low-res frames of worm embryos developing, and the model would fill in details and denoise, yielding high-quality segmentation of cells or tissues without needing high-end imaging at every timepoint. This approach of combining experimental and computational optimization is increasingly seen as a way to push biological imaging forward (analogous to how in astronomy, images are enhanced via algorithms). Thus, our foundation model not only analyzes data but effectively **augments the microscope's capabilities**. This could democratize certain experiments – labs with only basic microscopes might still perform complex analyses (like fine morphology or activity mapping) by leveraging the model's learned knowledge from high-end data.

- **Standardization and Reproducibility:** An often under-appreciated aspect of a foundation model is the standard it sets. By training on diverse datasets and being evaluated rigorously, our model could become a **community standard for worm image analysis**. This means results across labs can be compared more directly, as they would be using the same analysis platform rather than each lab writing their own code with slightly different thresholds

and criteria. For example, in aging research, labs measure worm lifespans or movement speed; if all use the same model for measuring motion, the results are more directly comparable and reproducible. We plan to release the model and an easy-to-use software package, which will encourage adoption. In the long run, this can lead to building even larger worm datasets (users contributing their data to further improve the model) and a positive feedback loop advancing *C. elegans* research. This aligns with trends in bioimage analysis toward centralized, well-validated tools[18].

Each of these examples demonstrates the **high-level impact** of our work. By solving the technical challenges in a unified way, we enable new science: higher throughput screens, finer phenotypic analyses, combined neural-behavior studies, and more. This is precisely the kind of advance expected of a *Nature Methods*-caliber contribution – not just an algorithm, but a platform that changes how experiments are done. Our proposal, if successful, will yield a tool that **empowers researchers to ask and answer questions that were previously impractical**, whether that's screening 1000 compounds for subtle behavioral effects or continuously monitoring neural circuit dynamics in a freely moving animal.

In conclusion, this research will deliver a foundation model tailored to *C. elegans* that is multi-task, multi-modal, and deployment-ready. We have outlined a comprehensive plan covering motivation, hypotheses, methodology, data, experiments, and applications. The work is ambitious yet grounded in strong preliminary evidence from related studies[15][19], and it addresses a clear gap in the current landscape of bioimage analysis. By adhering to rigorous academic standards and aiming for real-world utility, we expect the outcomes to meet the high bar of journals like *Nature Methods* or *Nature Machine Intelligence*. This project will not only advance the state of computational worm biology but also serve as a blueprint for developing foundation models in other biomedical domains, ultimately contributing to the broader goal of integrating AI deeply into biological research for accelerated discovery.

References: *(selected inline throughout the proposal)*

- Liu *et al.*, PLoS Comput Biol 2025 – deep learning-based worm detection & tracking framework[11][49].
- Guisnet & Hendricks, bioRxiv 2025 – TWARDIS foundation model pipeline for worms using SAM[19][10].
- Deserno & Bozek, Sci. Reports 2023 – WormSwin transformer for worm instance segmentation[29][9].
- Zhao *et al.*, Nature Methods 2025 – BiomedParse foundation model for multi-task biomedical image analysis[15][47].
- Zhang *et al.* 2023 – MobileSAM and KD-SAM for efficient Segment Anything Model distillation[23][30].

[1] [2] [3] [4] [6] [7] [11] [44] [49] [50] Automated C. elegans behavior analysis via deep learning-based detection and tracking | PLOS Computational Biology

https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1013707

[5] [8] [10] [19] [20] [26] [45] Large vision model framework for automated C. elegans analysis: From static morphometry to dynamic neural activity - PubMed

https://pubmed.ncbi.nlm.nih.gov/40894723/

[9] [12] [29] [37] [46] WormSwin: Instance segmentation of C. elegans using vision transformer | Scientific Reports

https://www.nature.com/articles/s41598-023-38213-7?error=cookies_not_supported&code=6fb1d705-b873-48bd-b997-0f50094b8313

[13] Automated Segmentation and Tracking of Caenorhabditis Elegans ...

https://imagescience.org/meijering/publications/1091/

[14] Segmentation and classification of two-channel C. elegans nucleus ...

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1817-3

[15] [16] [17] [18] [25] [47] A foundation model for joint segmentation, detection and recognition of biomedical objects across nine modalities | Nature Methods

https://www.nature.com/articles/s41592-024-02499-w?error=cookies_not_supported&code=06b70f20-f7c6-4eca-9ffa-fa46508584ab

[21] [22] [23] [24] [27] [30] [38] [39] [40] [48] Efficient Knowledge Distillation of SAM for Medical Image Segmentation

https://arxiv.org/html/2501.16740v1

[28] [33] [34] [35] [36] [41] [42] [43] Online Knowledge Distillation for Multi-Task Learning

https://openaccess.thecvf.com/content/WACV2023/papers/Jacob_Online_Knowledge_Distillation_for_Multi-Task_Learning_WACV_2023_paper.pdf

[31] [32] Overcoming data scarcity in biomedical imaging with a foundational multi-task model | Nature Computational Science

https://www.nature.com/articles/s43588-024-00662-z?error=cookies_not_supported&code=7caf5d12-7f33-4c5d-9de4-7351ea7daaba