Adapting Vision Transformers to Organoid Imaging

Yu Takagi
Nagoya Institute of Technology
The University of Tokyo
yutakagi322@gmail.com

Abstract

Organoids are an increasingly central scientific platform for drug screening, disease modeling, and regenerative medicine. Because organoid form is closely tied to function, AI-driven image analysis has drawn intense interest, yet every existing organoid imaging study evaluates representations on a single dataset and no work has quantitatively compared representations across the breadth of public organoid datasets. We close this gap with **OrgBench**, a benchmark of 12 backbones on 12 public organoid-imaging evaluation datasets covering detection, segmentation, classification, and temporal outcome prediction. The benchmark reveals a structured domain gap: task-family winners shift across the benchmark and parameter count is a poor predictor of transfer. Building on this, we introduce **OrgFM**, a bottleneck adapter that we continually pre-train on unlabeled organoid images while keeping a general-purpose vision transformer frozen. On the OrgBench task-family aggregates, OrgFM improves over its frozen base backbone. Layer- and feature-level interventions show that the downstream heads rely on a small set of late-layer morphology-sensitive features: the adapter sparsely supplements, rather than replaces, the base representation. Together, this is the first cross-task, cross-backbone study of organoid imaging at scale; the OrgBench/OrgFM pairing establishes that the parameter-efficient adapter direction is a promising path for organoid representations built on general-purpose backbones and shows where in the representation its gain resides.

1 Introduction

Organoids, self-organized three-dimensional tissue cultures that recapitulate the architecture and function of real organs [45, 17], are replacing animal experiments and accelerating drug discovery [42, 40], disease modeling [24], developmental biology [31], and regenerative medicine [6]. Because organoid form is closely tied to function, image analysis is the primary readout of these assays: bright-field and fluorescence images must be analyzed at scale to quantify morphology, counts, differentiation state, and temporal dynamics, and this image-analysis step is what now bottlenecks throughput. A decade of task-specific deep-learning pipelines has been developed for individual organoid assays (detection [21], instance segmentation and tracking [30, 25], semantic segmentation and morphometrics [12], phenotype classification [10, 2], and time-lapse outcome prediction [1]), each trained from scratch for one organ, one modality, and one annotation type; none transfers across the breadth of modern organoid biology.

A natural alternative is to start from a pre-trained vision foundation model (VFM). The past three years have produced a zoo of them (general self-supervised vision transformers (ViTs) [33, 38, 16], pathology-specific foundation models (FMs) trained on hematoxylin and eosin (H&E) stained whole-slide images [46, 28], cell-segmentation FMs trained on microscopy [34], and biomedical vision-language models [44, 36]), yet recent evaluations show that even strong VFMs transfer to medical imagery in a task- and modality-dependent way [39, 27], and no systematic evaluation exists

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

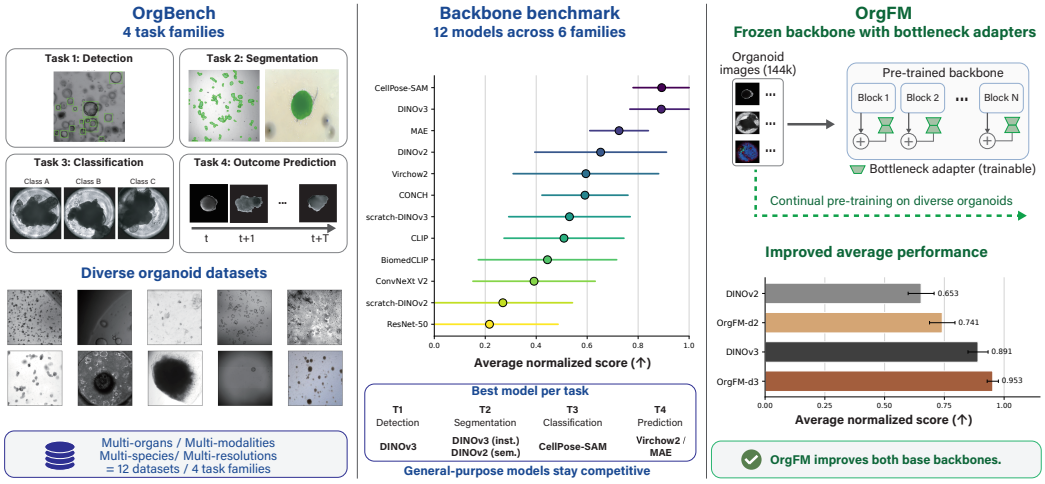

Figure 1: **Overview of OrgBench and OrgFM.** *Left:* OrgBench aggregates 12 public organoid-imaging evaluation datasets across detection, instance and semantic segmentation, classification, and temporal outcome prediction. *Middle:* A cross-comparison of 12 backbones shows that model scale alone does not close the organoid domain gap, and the per-task-family winner shifts across the benchmark. *Top right:* OrgFM continually pre-trains a small bottleneck adapter on a frozen ViT-B backbone (DINOv2 or DINOv3) using SimCLR on 144,339 unlabeled organoid images. *Bottom right:* Average normalized OrgBench score for the two base backbones and the two OrgFM-adapted backbones (mean $\pm$ SD over seeds, normalized as in Figure 4a). T4 (temporal outcome prediction) here is the early-prediction-window mean (timepoints 12–30, matching the highlighted band in Figure 4b), where the developmental outcome is hardest to predict.

for organoid imagery. In a domain where black-box failures waste experimental cycles, mechanistic-interpretability tools [3, 26] are a natural complement, but have not yet been applied to biomedical adaptation.

Treating organoid imaging as a use-inspired test case for domain adaptation of vision foundation models, we address three concrete questions a practitioner faces. First, which existing VFMs transfer best to organoid imaging? Second, can a small domain-specific intervention, with orders of magnitude fewer trainable parameters than full fine-tuning, improve transfer while leaving the base weights untouched? Third, can the resulting improvement be localized inside the representation rather than treated as a black-box gain?

Our contributions.

1. **OrgFM: parameter-efficient adaptation for organoid imaging** (Section 3.2). We show that training about 0.1% of a frozen VFM's parameters by self-supervised continual pre-training on unlabeled organoid images is enough to improve transfer. Our 92K-parameter bottleneck adapter (OrgFM) and a 74K-parameter LoRA both improve the normalized aggregate over OrgBench task families relative to the frozen base.
2. **OrgBench: a use-inspired evaluation suite for organoid-image representation learning** (Section 3.1). We benchmark four organoid imaging tasks (counting, morphometric segmentation, phenotype classification, and early temporal outcome prediction) under a unified frozen-backbone protocol on 12 public evaluation datasets. All artifacts are released at `https://github.com/yu-takagi/OrganoidFM`.
3. **Interventional analysis of adaptation** (Section 4.3). Using layer knockouts and sparse feature decompositions, we show that the downstream heads for some tasks rely on a small set of late-layer morphology-sensitive features: ablating them harms performance far more than count-matched random ablations.

Figure 1 summarizes the three components: the scale of OrgBench (left), the cross-backbone comparison it enables (middle), and the improvement produced by OrgFM (right).

Table 1: **OrgBench evaluation datasets.** 12 public organoid evaluation datasets organized into 4 task families (T1–T4). Two datasets contribute to two task families each (`tellu` to T1+T3, `afting` to T3+T4), and T2 is split into instance and semantic segmentation for evaluation, yielding *14 (task, dataset) evaluation cells* that we use to compute the per-task-family aggregates throughout the paper. T1 = detection (mAP@50); T2-inst = instance segmentation (mask mAP@50); T2-sem = semantic segmentation (mean IoU); T3 = classification (AUC); T4 = temporal outcome prediction (weighted F1). BF = bright-field; PhC = phase-contrast; Conf. = confocal fluorescence; CF = cystic fibrosis. #Img, #Train, and #Ann/Class are the counts of the variant we actually evaluate, not the raw upstream release; the two differ for datasets marked † (organoidnetdata, evaluation-time tiling; raw is 902 images) and ‡ (multiorg, post-tiling; raw is 411 large images / $\sim$60,147 annotations). All such cases are documented in Appendix A.3.

Dataset	Task(s)	Organ	Species	#Img	#Train	#Ann/Class	Modality
tellu [10]	T1, T3	intestine	mouse	840	681	23,065 / 4 cls	BF
orgaquant [21]	T1	intestine	human	1,742	1,467	14,139	BF
multiorg‡ [4]	T1	lung	hum./mou.	36,031	29,886	57,820	BF
orgasegment [25]	T2-inst	intestine (CF)	human	231	166	15,497 inst.	BF
organoidnetdata† [23]	T2-inst	PDAC	hum./mou.	912	642	65,225 inst.	PhC
organoid_ds_pdac [30]	T2-inst	PDAC	human	94	47	139 inst.	BF
brain1400 [37]	T2-sem	brain	human	1,407	1,011	n/a (sem.)	BF
cf_respiratory [8]	T2-sem	respiratory	human	827	708	n/a (sem.)	BF
organolabeler [20]	T2-sem	brain/enteroid	human	597	195	n/a (sem.)	BF / Conf.
opara§ [35]	T3	multi-organ	mixed	200	90	10 cls.	BF
organoid_classifier§ [2]	T3	hypothalamus	human	1,500	1,194	3 cls.	BF
afting [1]	T3, T4	retina	medaka	988	692	2+ cls. / 117K fr.	BF

2 Related Work

Organoid image analysis. Organoid imaging underpins studies of tissue heterogeneity, developmental dynamics, and therapeutic response across organs [29, 41, 32], but the analysis pipelines that consume it are built assay by assay (detection, instance / semantic segmentation, classification [11]) without shared backbones, splits, or evaluation protocols. This fragmentation makes it impossible to ask which representation should be used for a new assay, or whether domain-specific adaptation is worth its cost. A recent perspective highlights the potential of foundation models for stem cell-derived imaging [7]. Concurrent work introduces OrgLine, which combines detection and prompt-guided segmentation for organoid morphometry across multiple datasets [9]. To our knowledge, OrgBench is the first unified benchmark that compares frozen VFM representations across organoid detection, instance/semantic segmentation, phenotype classification, and temporal outcome prediction under a shared evaluation protocol.

Adapting vision foundation models to biomedical imaging. Biomedical vision foundation models span H&E pathology [46, 28], microscopy and cell segmentation [34], and biomedical image–text learning [44], each inheriting the biases of its corpus, and recent benchmarks confirm that even strong general-purpose ViTs transfer in a task- and modality-dependent way on broader medical imagery [27]: direct transfer to bright-field organoid microscopy is therefore not guaranteed. Closing the gap with domain-adaptive continual pre-training reliably helps in specialized domains [14, 13, 46, 28] but typically updates the full backbone, whereas adapter-based tuning [18, 19] keeps most weights fixed and the base representation intact. For biomedical workflows where labels are scarce and failure modes are costly, an improved aggregate metric is also not by itself sufficient grounds for deployment: where the change lives inside the representation matters, and sparse-dictionary decompositions [3, 26] provide a tractable tool for that question. Our approach, OrgFM, is adapter-based, and we go further by characterizing in detail what changes inside the adapted model relative to its frozen base representation.

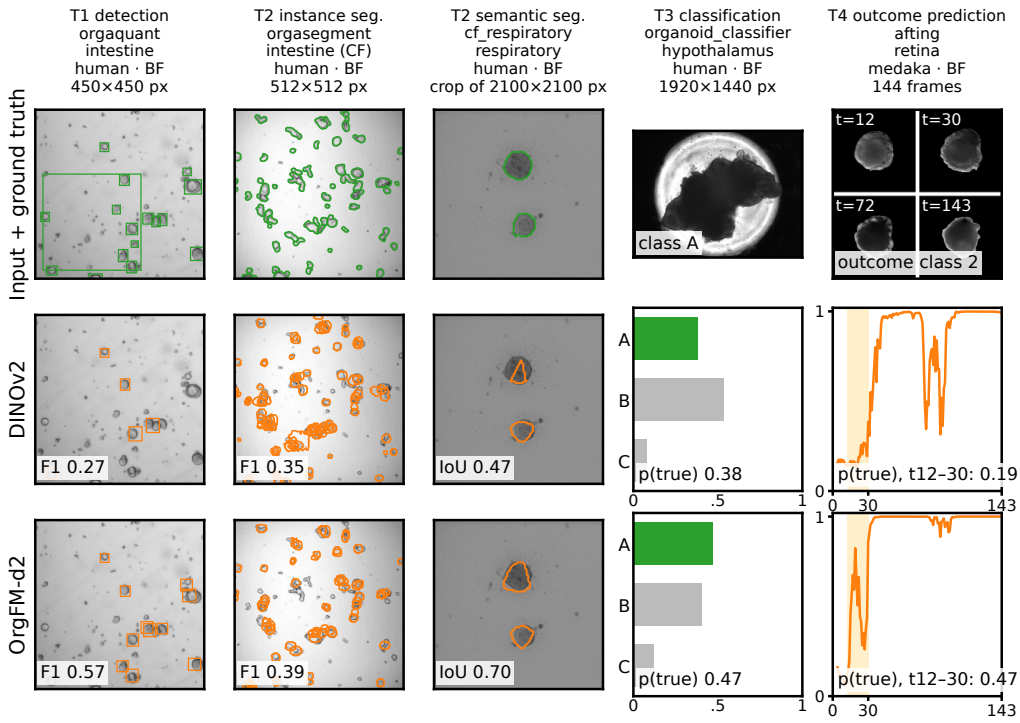

Figure 2: **Qualitative overview of OrgBench.** One test example per task family; each column gives the dataset, organ, species, modality, and image size. Rows show the input with the ground truth (green) and the predictions (orange) of the frozen DINOv2 backbone and of OrgFM-d2. Appendix A.2 adds, for every task, the examples with the largest gain and the largest drop from DINOv2 to OrgFM-d2, and the most uncertain examples.

3 Method

3.1 OrgBench construction

OrgBench aggregates 12 public organoid-imaging evaluation datasets (Table 1) into four task families: Task 1 (T1), detection; Task 2 (T2), segmentation (split into instance and semantic); Task 3 (T3), classification; and Task 4 (T4), temporal outcome prediction. Together these cover intestine, lung, brain, pancreas (pancreatic ductal adenocarcinoma, PDAC), retina, respiratory tract, and multi-organ collections in human, mouse, and medaka, spanning bright-field, phase-contrast, and confocal-fluorescence modalities. A thirteenth public organoid dataset, `organoid_ds_mouse` (n_{train}=10), is too small for stable downstream evaluation and is included only as a pre-training source (Appendix A.3, Table 7); it is not part of the OrgBench evaluation suite. Figure 2 shows a test example for one dataset of each task family.

OrgBench evaluates **12 backbones** (listed as architecture, parameter count): DINOv2 [33] (ViT-B/14, 86M), DINOv3 [38] (ViT-B/16, 86M), MAE [16] (ViT-B/16, 86M), CLIP [36] (ViT-B/16, 86M), BiomedCLIP [44] (ViT-B/16, 86M), CONCH [28] (ViT-B/16, 86M), Virchow2 [46] (ViT-H/14, 632M), CellPose-SAM [34] (ViT-L/8 from the Segment Anything Model, 308M), ResNet-50 [15] (CNN, 25M), ConvNeXt V2-B [43] (CNN, 89M), and two scratch-initialized ViT-B controls (scratch-dinov2: ViT-B/14, 86M; scratch-dinov3: ViT-B/16, 86M).

Evaluation protocol. All backbones are evaluated under a unified frozen-backbone plus lightweight-head protocol: Mask-/Faster-R-CNN with a ViTDet-style neck for T1 and T2-inst, a lightweight upsampling decoder for T2-sem, a linear probe on frozen image features for T3, and a single-layer transformer over per-frame embeddings for T4. Head architectures are deliberately simple so that OrgBench primarily measures representation transfer rather than task-specific engineering. Each

task family's head is trained independently on the same frozen features, so that differences between backbones reflect transfer from a shared representation rather than interactions between tasks. We leave joint multi-task training to future work. All datasets, splits, checkpoints, and pre-training and evaluation code will be released; full training details are documented in Appendix A.4.

3.2 OrgFM: bottleneck adapter trained with SimCLR

Architecture, objective, and hyperparameters. OrgFM is a bottleneck adapter [18] injected into frozen ViT-B backbones, yielding two variants: OrgFM-d2 (DINOv2 base) and OrgFM-d3 (DINOv3 base). For each of the 12 transformer blocks we insert two bottleneck adapters (one after the attention sublayer and one after the MLP sublayer), yielding 24 adapter modules in total with bottleneck width 2 ($\sim$92K trainable parameters); both variants share this architecture. A projection head $768 \to 128 \to 128$ is trained jointly with the adapter and discarded after pre-training; only the adapter and the projection head receive gradients, and the backbone remains frozen throughout training. We optimize a SimCLR [5] InfoNCE objective with a SimCLR-style augmentation set (Appendix A.3). Optimization uses AdamW with a constant learning rate, and performance saturates by 26 epochs of adapter-only training (5,000 samples per epoch, thus corresponding roughly one full pass over the pre-training pool; Figure 4d). We confirm that the OrgFM-over-base improvement is robust across the hyperparameters rather than restricted to a particular configuration. The two variants converge on the same architectural choices (bottleneck width 2, projector 128, $\sqrt{N}$-proportional sampling, 26 epochs) but settle on different optimization hyperparameters (learning rate and batch size); the full grid and chosen configurations are in Appendix A.6.

Pre-training data. We curate an organoid-imaging pre-training pool of **144,339** images aggregated from all 13 public organoid datasets (12 sources, as the two OrganoID subsets are pooled), combining each dataset's train and val splits and excluding all benchmark test images. The pool is dominated by the afting time-lapse retinal dataset ($\sim 74\%$ of images, reflecting what is publicly available rather than a deliberate balance); the remaining 37,821 images cover intestine, lung, brain, pancreas, respiratory tract, hypothalamus, and multi-organ collections across human, mouse, and medaka. All pre-training is self-supervised and labels are used only for downstream-task evaluation. Per-dataset composition and the dataset-specific preprocessing choices that align resolution between pre-training and benchmark evaluation are documented in Appendix A.3, Table 7. A leave-one-source-out ablation (Figure 4c) further verifies that no single source is individually critical for the dense or temporal tasks.

3.3 Interpretation methods

Beyond measuring that OrgFM improves tasks, we ask where and how the adaptation acts inside the backbone, and whether the downstream heads rely on the localized signal. We use two complementary lines of analysis, each pairing a coarse probe with a targeted intervention; the analyses are combined into a single interpretation in Section 4.3, and full protocols, hyperparameters, and validation controls are in Appendix A.7.

(i) Layer-level interpretation (Figure 5a–d). We first ask *at which depth* the adaptation lives. We extract patch tokens after each of the 12 transformer blocks of the ViT-B backbone and (a) compare layer-wise centered kernel alignment (CKA) [22] between base and adapted backbones; (b) train a logistic-regression linear probe on each layer's class (CLS) token for the four T3 datasets to locate the layer carrying the most task-relevant information; (c) and regress low-level morphology primitives (size, brightness, eccentricity, circularity) from frozen features on the same four T3 datasets to measure how linearly those concepts are encoded. To test the resulting layer localization by intervention, we additionally run a *layer-level knockout*: for each transformer block $\ell \in \{0, \ldots, 11\}$ we skip both adapter modules inserted in that block by setting their up-projection weights to zero, leaving all adapters in the other blocks intact, measure the downstream task change, and contrast individual-block knockouts with simultaneous knockout of all 12 blocks.

(ii) Feature-level interpretation with a Crosscoder (Figure 5e–h). We then ask *which features* carry the adaptation. Crosscoders were introduced in mechanistic interpretability as a tool for examining how a model's features change between two related checkpoints (typically a base model and its fine-tuned counterpart) by jointly decomposing aligned activations from both models with a single sparse-feature dictionary, so that the dictionary itself separates what the two models share from what

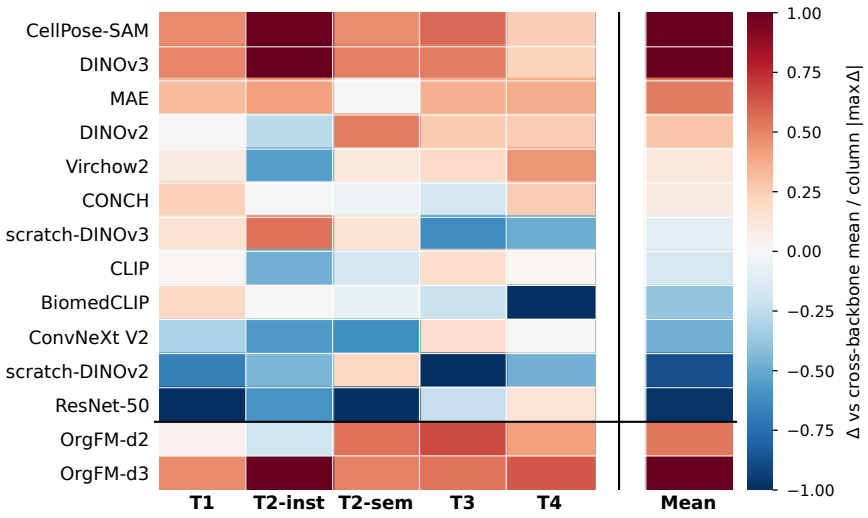

Figure 3: **OrgBench landscape.** Per-task-family relative performance gain (% vs. the cross-backbone mean for that task family) for all 12 OrgBench backbones across tasks, with OrgFM-d2 and OrgFM-d3 below the line for reference. Each cell is computed by first averaging the 3-seed mean scores over the constituent datasets of that task family per backbone, then expressing it as percent improvement over the mean of the 12 backbones for that family. Each column is then independently rescaled by its own $|\max|$ over the 12 backbones so the colormap is comparable across task families. Note that OrgFM-d2 and OrgFM-d3 enter neither the mean nor the maximum. The backbones are sorted by the rightmost *Mean* column, which averages the five columns to its left and is rescaled in the same way. Raw scores are reported in Appendix A.1, Table 2.

each represents idiosyncratically [26]. We adopt this exact use case here, treating the frozen base backbone and the OrgFM-adapted backbone as the two checkpoints. Concretely, we fit a Crosscoder, a top-k sparse autoencoder applied jointly to base and adapted activations, that decomposes aligned patch-token pairs $(h_\ell^{\text{base}}, h_\ell^{\text{orgfm}})$ at layer ℓ into three disjoint feature dictionaries:

- *shared* features that reconstruct both activations;
- *OrgFM-specific* features used only by the adapted model;
- *base-specific* features used only by the backbone.

We train separate Crosscoders at L3 (early), L7 (middle), and L11 (output of the last transformer block) on 5,000 images sampled from the organoid pre-training pool, decomposing patch-token activations only (CLS is excluded from the decomposition), and using a canonical $8\times$-expansion dictionary with top-K sparsity ($K{=}128$ active features per input) for both the qualitative split and the knockout study. We report alive-feature statistics because deeper layers exhibit high dead-feature rates, and validate the decomposition with a *self–self* control that feeds the same model into both inputs (where a faithful decomposition should yield near-zero specific-feature counts) and a $4\times/8\times$ dictionary-size sensitivity sweep. To test whether the downstream heads rely on the OrgFM-specific subset, we add feature-level knockouts at L11. We compare against a size-matched random knockout of alive features and a control that re-injects the un-zeroed reconstruction (*recon*). Full protocols are deferred to Appendix A.7.

4 Results

The results are organized in three parts: OrgBench quantifies the organoid domain gap across 12 backbones (Section 4.1); continual pre-training of a bottleneck adapter narrows it, improving the cross-task aggregate for both base backbones (Section 4.2); and feature-level analysis with targeted knockouts identifies late-layer features on which the downstream heads rely (Section 4.3). Figure 1 in the introduction provides a one-figure summary.

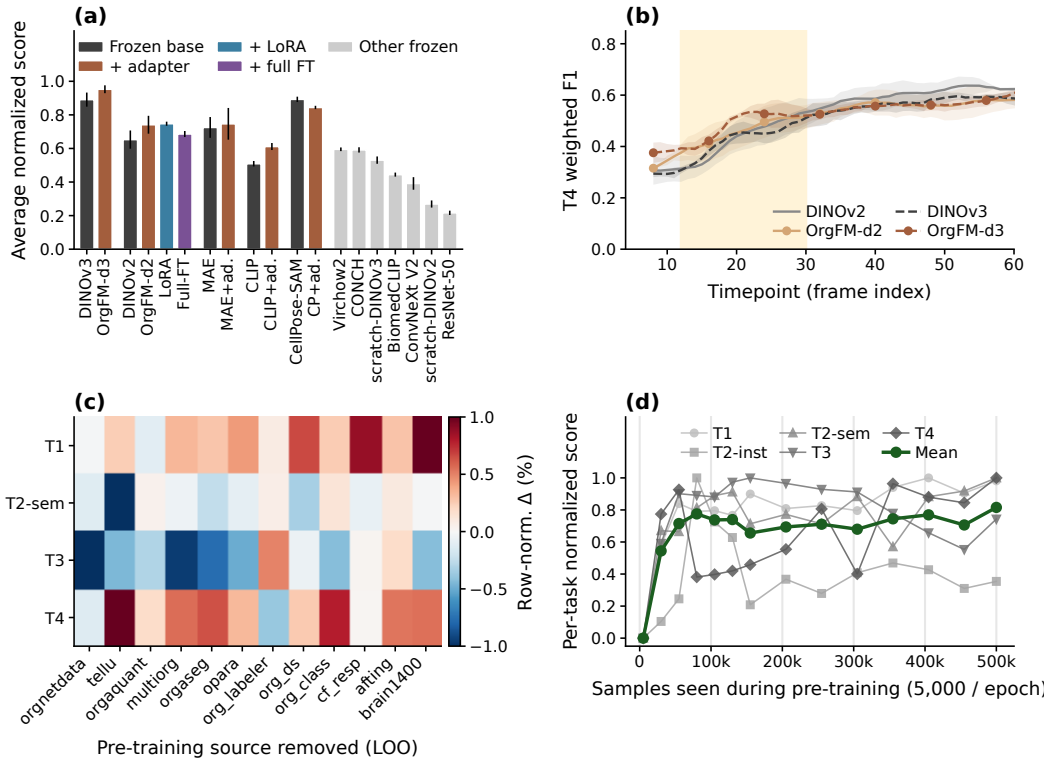

Figure 4: **OrgFM effect across task families.** (**a**) Average OrgBench score over all task families, with each task family min–max normalized across them; mean ± SD over seeds. (**b**) T4 per-timepoint weighted F1, 5-frame rolling mean; the yellow band marks the early window (timepoints 12–30) where OrgFM's gain is concentrated. (**c**) Leave-one-source-out heatmap: relative change in each task-family mean metric when each pre-training source is removed; rows row-normalized with |max| per task. (**d**) Aggregate sample-budget saturation under the canonical OrgFM-d2 sampler (5,000 samples / epoch). Each task family is min–max normalized within the 14-checkpoint sweep (gray lines, one marker shape per family) and the mean across T1–T4 is shown in green.

4.1 OrgBench quantifies the organoid domain gap

Figure 3 shows the cross-backbone landscape as a heatmap of relative performance gain per task family; raw scores for all backbones are tabulated in Appendix A.1, Table 2. We aggregate all tasks and datasets cells (Table 1) when computing per-task-family scores. Two observations stand out:

Generalist VFMs are strong starting points. The strongest backbones on OrgBench are general self-supervised ViTs (DINOv3, MAE, DINOv2) together with CellPose-SAM, the only microscopy-specific FM in the panel. Pre-training tuned to nearby biomedical modalities does not automatically help: the largest pathology FM (632M) is at or below several 86M general ViTs across task families other than T4, and biomedical vision-language models trained on broader medical text-image pairs trail DINOv3 and MAE on the dense tasks. Within OrgBench, neither sheer parameter count nor target-domain biomedical pre-training is sufficient: a general-purpose self-supervised ViT often transfers as well as or better than the supposedly closer biomedical FMs.

No single backbone dominates all task families. The per-task-family winner shifts across T1–T4: detection (T1) and instance segmentation (T2-inst) are led by DINOv3, semantic segmentation (T2-sem) is led by DINOv2, classification (T3) is led by CellPose-SAM, and temporal outcome prediction (T4) is closely contested. A practitioner who selects the best backbone separately for each task would therefore need to maintain several parallel pipelines, which motivates a single, low-cost adaptation strategy that lifts a strong general-purpose backbone toward competitive performance across task families (Section 4.2).

4.2 Continual pre-training improves and complements base backbones

Figure 4a reports scores for all 19 configurations, with each adapted configuration next to its frozen base. Two findings stand out. First, on the normalized aggregate over OrgBench task families each OrgFM variant improves over its corresponding frozen base (full per-family numbers in Appendix Table 3). Second, simple output-level ensemble that include DINOv3 and OrgFM-d3 outperforms both of its members on every task family of Table 9 (Appendix A.5), suggesting that the adapted backbone carries signal that its base lacks. Section 4.3 pins down what specifically the adaptation changed and where in the backbone it lives.

Beyond OrgFM, we compared the adapter with two other ways of adapting DINOv2 during continual pre-training and applied the same adapter to three other backbones (Appendix A.4). In the average normalized score over all 19 configurations, LoRA is about as effective as the adapter, whereas full fine-tuning of all parameters gives a smaller gain. The adapter also improves CLIP and MAE but lowers CellPose-SAM, suggesting that it helps more when the frozen backbone is weaker or more general-purpose than when it is already specialized for cell microscopy. Appendix Tables 3 and 5 list the raw scores and the seed-paired differences by dataset, task family, and species. Absolute instance-segmentation performance varies widely across datasets, with the best mask mAP@50 ranging from 0.068 to 0.556.

The aggregate T4 score in Figure 4a collapses across timepoints; resolving it per timepoint (Figure 4b) reveals a sharper structure: OrgFM's benefit is concentrated at the early timepoints where the developmental outcome is still hardest to predict, with a large per-window gain over the base backbone. At late timepoints, when morphology is visually obvious, all four models converge to within a few points of each other. The adapter therefore obtains early predictive access to the outcome, useful for any setting in which a decision must be made before the organoid has fully differentiated, rather than uniform improvement across the trajectory.

The leave-one-source-out (LOO) heatmap (Figure 4c) shows that T1, T2-sem, T3, and T4 are all robust to removing any single pre-training source: OrgFM does not hinge on afting despite its dominant share of the pool, and removing it mainly lowers T3 on afting itself (raw changes in Appendix Table 10). Removing organoidnetdata, the only phase-contrast source, gives one of the two largest mean drops in T3 accuracy; we hypothesize that this reflects the loss of the only phase-contrast images. Note that here we omit T2-inst because mask mAP@50 is too low on these datasets to give a stable LOO signal.

The aggregate normalized score across T1–T4 (Figure 4d) reaches its first plateau early in training and remains within a narrow band thereafter; additional pre-training does not improve all tasks monotonically. We report all numbers at a post-plateau epoch-25 checkpoint. Together with the leave-one-out result above and the hyperparameter robustness in Appendix A.6, this means the headline OrgFM scores do not rely on a specific tuned configuration.

4.3 Interpretation: where and how the adapter acts

We combine the two analyses of Section 3.3 into a single mechanistic picture (Figure 5). The top row (**a**–**d**) is a layer-level interpretation that asks *where* the adaptation lives inside the backbone, and the bottom row (**e**–**h**) is a feature-level interpretation built from a Crosscoder that asks *what kind of features* carry the adaptation at that depth. We analyze the DINOv2/OrgFM-d2 pair, because OrgFM-d2 improves every task-family mean over DINOv2 and thus offers an internally consistent adaptation effect to dissect.

A. Layer-level interpretation. Layer-wise linear CKA between the base and adapted backbones (Figure 5a) stays high in early layers and drops in late layers, indicating that the representational divergence between base and adapted backbones is concentrated in the late layers of the network. Layer-wise T3 linear probes (Figure 5b) show that the base backbone peaks in mid-network and degrades by L11, whereas OrgFM-d2 keeps L11 close to its peak. Best-layer concept probes for organoid morphology primitives (Figure 5c) yield consistent R^2 gains under OrgFM-d2 on size, brightness, and eccentricity. Finally, replacing each adapter block in turn with the identity (Figure 5d) suggests the T3 gain is distributed across depth rather than concentrated in any single block. Together, the representational and interventional pictures do not align one-to-one, motivating the following feature-level decomposition.

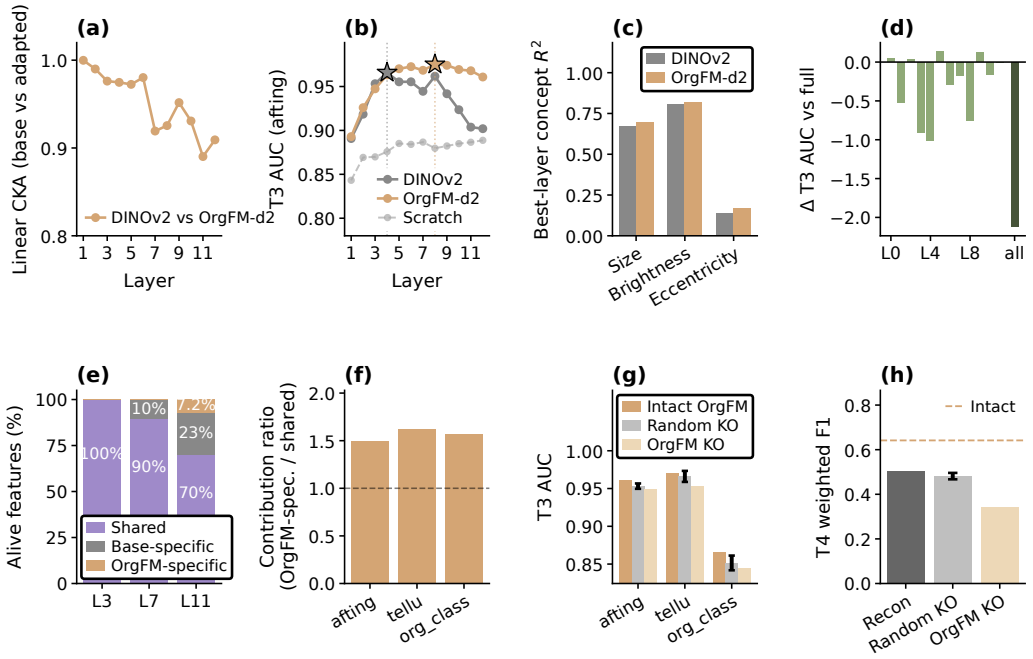

Figure 5: **Mechanistic picture of OrgFM (DINOv2 base) adaptation.** Top row (**a**–**d**): layer-level interpretation. Bottom row (**e**–**h**): feature-level interpretation (Crosscoder); see Section 4.3 for analysis and Appendix A.7 for full protocols. (**a**) Layer-wise linear CKA between base and adapted backbones on a held-out organoid pool. (**b**) Per-layer T3 AUC of a logistic-regression probe on the layer's CLS token (afting), with $\star$ marking the best layer. (**c**) Best-layer R^2 for four organoid morphology primitives (size, brightness, eccentricity, circularity), base vs adapted. (**d**) Per-block adapter knockout: ΔT3 AUC when one adapter block is replaced with the identity, plus the all-12-block knockout (no head retraining). (**e**) Fraction of alive features per Crosscoder category at L3, L7, L11. (**f**) Per-feature T3 head weight, read out without running the model: each feature's attribution a_i is the mean (over classes) absolute weight that the trained T3 head places on its OrgFM-side decoder direction; the bar plots the ratio of the within-category mean attribution, $\bar{a}_{\text{OrgFM-specific}}/\bar{a}_{\text{shared}}$, at L11 for three T3 datasets, so a value above the dashed unit line means OrgFM-specific features carry more per-feature head weight than shared features (formal definition in Appendix A.7). (**g**) T3 subspace ablation: AUC after projecting the L11 OrgFM-specific decoder directions out of the CLS embedding and refitting the linear head; size-matched random control from alive shared features and the intact OrgFM as a reference. (**h**) T4 forward intervention: weighted F1 (average across all timepoints) after replacing L11 patch tokens with the Crosscoder reconstruction with OrgFM-specific columns zeroed and feeding them through the pre-trained T4 transformer without retraining. Reconstruction control re-injects the un-zeroed reconstruction; intact OrgFM as a horizontal reference.

B. Feature-level interpretation via Crosscoder. Crosscoder decomposition (Figure 5e) shows that early and middle activations are mostly shared between base and adapted models, while L11 contains a small OrgFM-specific subset and a comparable base-specific subset weakened by adaptation. Composing the trained T3 head with the OrgFM-side decoder gives, per feature, the weight the head places on it without ever activating the model; under this static read-out, OrgFM-specific features carry roughly twice the per-feature head weight of shared features at L11 across the T3 datasets (Figure 5f, bar above the dashed unit line). Some of them also activate on organoid boundaries and internal structures in image-space visualizations (Appendix A.7, Figure 12), and as a group they track morphology primitives at a similarly elevated rate. We confirm that self–self Crosscoders yield near-zero specific-feature counts and the decomposition is stable across dictionary-size sweeps. Furthermore, ablating only the L11 OrgFM-specific subset — by projecting these directions out of the CLS embedding for T3 (and refitting the linear head) and by replacing the patch-token activations with the zeroed Crosscoder reconstruction for T4 (and feeding them through the pre-trained transformer without retraining) — consistently degrades downstream accuracy, while a size-matched random

knockout of the same number of alive features causes a substantially smaller drop (Figure 5g, h). For the DINOv2/OrgFM-d2 pair on T3 and T4, the OrgFM-specific subset is therefore small but disproportionately important for downstream task performance: adaptation works by adding a few late-layer, morphology-sensitive features on top of an otherwise unchanged backbone.

5 Discussion

OrgBench is a cross-comparison of vision representations for organoid imaging. Across 12 backbones and four task families, no single backbone is uniformly best and parameter count alone is a poor predictor of transfer. Continual pre-training of a small bottleneck adapter on top of a frozen general-purpose ViT (OrgFM) improves the normalized aggregate over OrgBench task families relative to its frozen base (DINOv2 or DINOv3), as does a LoRA of similar size, and both gain more than full fine-tuning; ensembles that mix base and adapted backbones further improve every task family over each constituent. For the DINOv2/OrgFM-d2 pair, layer- and feature-level knockouts on T3 and T4 then trace these gains to a small set of deep-layer features whose ablation disproportionately damages downstream performance.

The OrgBench ranking offers a perhaps surprising message about which pre-training distributions transfer to organoid imagery: general self-supervised ViTs (DINOv2, DINOv3, MAE) are consistently among the strongest starting points, while pre-training tuned to other biomedical modalities (H&E pathology, natural image-text pairs) underperforms at the same or larger parameter budget. This echoes the broader picture flagged in the introduction and related work [39, 27]: which VFM transfers best to a new domain is not predictable from parameter count or apparent domain proximity, and has to be measured empirically per task family.

The OrgFM gain shows up consistently across multiple complementary angles rather than in any single aggregate score. First, behavior at the level of model combinations: each OrgFM variant individually improves over its frozen base on the normalized cross-task-family aggregate, and output-level ensembles that mix base and adapted backbones improve further on every task family — indicating that the adapted and base backbones carry non-redundant information. Second, behavior resolved along time: the aggregate T4 score collapses across timepoints, but resolving it per timepoint shows that the gain concentrates at the early windows where the developmental outcome is hardest to predict from morphology alone. Third, robustness: the T3 gain does not hinge on any single pre-training source and saturates early in pre-training rather than tracking the epoch budget.

The mechanistic analysis tells a consistent story at the representation level: OrgFM acts on top of the base backbone rather than overwriting it. CKA and linear probes place the change in deep layers, where the base backbone's representation is most degraded relative to its mid-network peak; concept probes for organoid morphology primitives improve at those depths, and ablating individual adapter blocks shows the T3 gain is distributed across depth rather than concentrated in any single block. At the late layer where this contribution lands, a Crosscoder isolates a small subset of features that are present in the adapted model but absent in the base, and the trained downstream head places disproportionately more weight on a typical such feature than on a typical shared one. Ablating only this OrgFM-specific subset consistently degrades T3 and T4 performance, identifying these features as carriers of the gain for these tasks.

Acknowledgments and Disclosure of Funding

Y.T. was supported by JST PRESTO (JPMJPR23I6), JST CREST (JPMJCR2574) and JSPS KAKENHI (Grant Numbers JP19H05725 and JP24K02999).

References

[1] Cassian Afting, Norin Bhatti, Christina Schlagheck, Encarnación Sánchez Salvador, Laura Herrera-Astorga, Rashi Agarwal, Risa Suzuki, Nicolaj Hackert, Hanns-Martin Lorenz, Lucie Zilova, Joachim Wittbrodt, and Tarik Exner. A deep learning-based computational pipeline predicts developmental outcome in retinal organoids. *PLOS Biology*, 24(1):e3003597, 2026. doi: 10.1371/journal.pbio.3003597. URL `https://doi.org/10.1371/journal.pbio.3003597`.

[2] Tomoyoshi Asano, Hidetaka Suga, Hirohiko Niioka, Hiroshi Yukawa, Mayu Sakakibara, Shiori Taga, Mika Soen, Tsutomu Miwata, Hiroo Sasaki, Tomomi Seki, Saki Hasegawa, Sou Murakami, Masatoshi Abe, Yoshinori Yasuda, Takashi Miyata, Tomoko Kobayashi, Mariko Sugiyama, Takeshi Onoue, Daisuke Hagiwara, Shintaro Iwama, Yoshinobu Baba, and Hiroshi Arima. A deep learning approach to predict differentiation outcomes in hypothalamic–pituitary organoids. *Communications Biology*, 7:1468, 2024. doi: 10.1038/s42003-024-07109-1. URL https://doi.org/10.1038/s42003-024-07109-1.

[3] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[4] Christina Bukas, Harshavardhan Subramanian, Fenja See, Carina Steinchen, Ivan Ezhov, Gowtham Boosarpu, Sara Asgharpour, Gerald Burgstaller, Mareike Lehmann, Florian Kofler, and Marie Piraud. MultiOrg: A multi-rater organoid-detection dataset. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, pages 95808–95839, 2024. doi: 10.52202/079017-3036. URL https://doi.org/10.52202/079017-3036.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

[6] Woo Hee Choi, Dong Hyuck Bae, and Jongman Yoo. Current status and prospects of organoid-based regenerative medicine. *BMB Reports*, 56(1):10–14, 2023. doi: 10.5483/BMBRep.2022-0195. URL https://doi.org/10.5483/BMBRep.2022-0195.

[7] Luca Deininger, Paolo Caldarelli, Magdalena Zernicka-Goetz, and Ralf Mikut. From pixels to patterns: the AI revolution in stem cell-derived models. *Nature Methods*, 23:1710–1723, 2026. doi: 10.1038/s41592-026-03202-x. URL https://doi.org/10.1038/s41592-026-03202-x.

[8] Anna Demchenko, Maxim Balyasin, Elena Kondratyeva, Tatiana Kyian, Alyona Sorokina, Marina Loguinova, and Svetlana Smirnikhina. A semi-automated algorithm for image analysis of respiratory organoids. *PLOS Computational Biology*, 21(10):e1013589, 2025. doi: 10.1371/journal.pcbi.1013589. URL https://doi.org/10.1371/journal.pcbi.1013589.

[9] Xun Deng, Xinyu Hao, Thomas Herget, Mei Gao, Mathias Winkel, Feng Tan, Lun Hu, and Pengwei Hu. OrgLine: A versatile pipeline for organoid morphometry using detector-guided prompts. *Cell Reports Methods*, 6:101482, 2026. doi: 10.1016/j.crmeth.2026.101482. URL https://doi.org/10.1016/j.crmeth.2026.101482.

[10] Eva Domènech-Moreno, Anders Brandt, Toni T. Lemmetyinen, Linnea Wartiovaara, Tomi P. Mäkelä, and Saara Ollila. Tellu—an object-detector algorithm for automatic classification of intestinal organoids. *Disease Models & Mechanisms*, 16(3):dmm049756, 2023. doi: 10.1242/dmm.049756. URL https://doi.org/10.1242/dmm.049756. Associated dataset DOI: 10.5281/zenodo.6768583.

[11] Xuan Du, Zhenzhen Chen, Qi Li, Sheng Yang, Lulu Jiang, Yi Yang, Yanhui Li, and Zhongze Gu. Organoids revealed: Morphological analysis of the profound next generation in-vitro model with artificial intelligence. *Bio-Design and Manufacturing*, 6(3):319–339, 2023. doi: 10.1007/s42242-022-00226-y. URL https://doi.org/10.1007/s42242-022-00226-y.

[12] Nicola Gritti, Jia Le Lim, Kerim Anlaş, David Oriola, Mallica Pandya, Germaine Aalderink, Guillermo Martinez Ara, and Vikas Trivedi. MOrgAna: Accessible quantitative analysis of organoids with machine learning. *Development*, 148(18):dev199611, 2021. doi: 10.1242/dev.199611. URL https://doi.org/10.1242/dev.199611.

[13] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23, 2021. doi: 10.1145/3458754. URL https://doi.org/10.1145/3458754.

[14] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8342–8360, 2020. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[17] Moritz Hofer and Matthias P. Lutolf. Engineering organoids. *Nature Reviews Materials*, 6(5):402–420, 2021. doi: 10.1038/s41578-021-00279-y. URL https://doi.org/10.1038/s41578-021-00279-y.

[18] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*, 2019.

[19] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[20] Burak Kahveci, Elifsu Polatli, Yalin Bastanlar, and Sinan Guven. OrganoLabeler: A quick and accurate annotation tool for organoid images. *ACS Omega*, 9(46):46117–46128, 2024. doi: 10.1021/acsomega.4c06450. URL https://doi.org/10.1021/acsomega.4c06450.

[21] Timothy Kassis, Victor Hernandez-Gordillo, Ronit Langer, and Linda G. Griffith. OrgaQuant: Human intestinal organoid localization and quantification using deep convolutional neural networks. *Scientific Reports*, 9:12479, 2019. doi: 10.1038/s41598-019-48874-y. URL https://doi.org/10.1038/s41598-019-48874-y.

[22] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[23] Ajinkya Kulkarni, Nathalia Ferreira, Riccardo Scodellaro, et al. A curated cell life imaging dataset of immune-enriched pancreatic cancer organoids with pre-trained AI models. *Scientific Data*, 11:820, 2024. doi: 10.1038/s41597-024-03631-3. URL https://doi.org/10.1038/s41597-024-03631-3.

[24] Madeline A. Lancaster and Meritxell Huch. Disease modelling in human organoids. *Disease Models & Mechanisms*, 12(7):dmm039347, 2019. doi: 10.1242/dmm.039347. URL https://doi.org/10.1242/dmm.039347.

[25] Juliet W. Lefferts, Suzanne Kroes, Matthew B. Smith, et al. OrgaSegment: Deep-learning based organoid segmentation to quantify CFTR dependent fluid secretion. *Communications Biology*, 7:319, 2024. doi: 10.1038/s42003-024-05966-4. URL https://doi.org/10.1038/s42003-024-05966-4.

[26] Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Sparse crosscoders for cross-layer features and model diffing. Transformer Circuits Thread, 2024. URL https://transformer-circuits.pub/2024/crosscoders/index.html.

[27] Che Liu, Yinda Chen, Haoyuan Shi, Jinpeng Lu, Bailiang Jian, Jiazhen Pan, Linghan Cai, Jiayi Wang, Jieming Yu, Ziqi Gao, Xiaoran Zhang, Long Bai, Yundi Zhang, Jun Li, Cosmin I. Bercea, Cheng Ouyang, Chen Chen, Zhiwei Xiong, Benedikt Wiestler, Christian Wachinger, James S. Duncan, Daniel Rueckert, Wenjia Bai, and Rossella Arcucci. Does DINOv3 set a new medical vision standard? Benchmarking 2D and 3D classification, segmentation, and registration. *arXiv preprint arXiv:2509.06467*, 2025. URL https://arxiv.org/abs/2509.06467.

[28] Ming Y. Lu, Bowen Chen, Drew F. K. Williamson, Richard J. Chen, Ivy Liang, Tong Ding, Guillaume Jaume, Igor Odintsov, Long Phi Le, Georg Gerber, Anil V. Parwani, Andrew Zhang, and Faisal Mahmood. A visual-language foundation model for computational pathology. *Nature Medicine*, 30(3):863–874, 2024. doi: 10.1038/s41591-024-02856-4. URL https://doi.org/10.1038/s41591-024-02856-4.

[29] Ilya Lukonin, Denise Serra, Ludivine Challet Meylan, Kiara Volkmann, Janos Baaten, Rui Zhao, Shelly Meeusen, Karin Colman, Fabian Maurer, Michael B. Stadler, Jeremy Jenkins, and Prisca Liberali. Phenotypic landscape of intestinal organoid regeneration. *Nature*, 586(7828):275–280, 2020. doi: 10.1038/s41586-020-2776-9. URL https://doi.org/10.1038/s41586-020-2776-9.

[30] Jonathan M. Matthews, Brooke Schuster, Sara Saheb Kashaf, Ping Liu, Rakefet Ben-Yishay, Dana Ishay-Ronen, Evgeny Izumchenko, Le Shen, Christopher R. Weber, Margaret Bielski, Sonia S. Kupfer, Mustafa Bilgic, Andrey Rzhetsky, and Savaş Tay. OrganoID: A versatile deep learning platform for tracking and analysis of single-organoid dynamics. *PLOS Computational Biology*, 18(11):e1010584, 2022. doi: 10.1371/journal.pcbi.1010584. URL https://doi.org/10.1371/journal.pcbi.1010584.

[31] Heather A. McCauley and James M. Wells. Pluripotent stem cell-derived organoids: Using principles of developmental biology to grow human tissues in a dish. *Development*, 144(6):958–962, 2017. doi: 10.1242/dev.140731. URL https://doi.org/10.1242/dev.140731.

[32] Michelle O'Hara-Wright and Anai Gonzalez-Cordero. Retinal organoids: A window into human retinal development. *Development*, 147(24):dev189746, 2020. doi: 10.1242/dev.189746. URL https://doi.org/10.1242/dev.189746.

[33] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[34] Marius Pachitariu, Michael Rariden, and Carsen Stringer. Cellpose-SAM: Superhuman generalization for cellular segmentation. *bioRxiv*, 2025. doi: 10.1101/2025.04.28.651001. URL https://doi.org/10.1101/2025.04.28.651001.

[35] Jannik Presberger, Rashmiparvathi Keshara, David Stein, Yung Hae Kim, Anne Grapin-Botton, and Bjoern Andres. Correlation clustering of organoid images. In *Pattern Recognition – 46th DAGM German Conference, GCPR 2024*, volume 15297 of *Lecture Notes in Computer Science*. Springer, 2024. doi: 10.1007/978-3-031-85181-0_15. URL https://doi.org/10.1007/978-3-031-85181-0_15. Dataset hosted at OPARA (TU Dresden data repository), DOI: 10.25532/OPARA-598.

[36] 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 *International Conference on Machine Learning (ICML)*, 2021.

[37] Julian Schröter, Luca Deininger, Blaz Lupse, et al. A large and diverse brain organoid dataset of 1,400 cross-laboratory images of 64 trackable brain organoids. *Scientific Data*, 11(1):514, 2024. doi: 10.1038/s41597-024-03330-z. URL https://doi.org/10.1038/s41597-024-03330-z.

[38] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. doi: 10.48550/arXiv.2508.10104. URL https://arxiv.org/abs/2508.10104.

[39] Hamid R Tizhoosh. Beyond the failures: Rethinking foundation models in pathology. *arXiv preprint arXiv:2510.23807*, 2025.

[40] Georgios Vlachogiannis, Somaieh Hedayat, Alexandra Vatsiou, et al. Patient-derived organoids model treatment response of metastatic gastrointestinal cancers. *Science*, 359(6378):920–926, 2018. doi: 10.1126/science.aao2774. URL https://doi.org/10.1126/science.aao2774.

[41] Philipp Wahle, Giovanna Brancati, Christoph Harmel, Zhisong He, Gabriele Gut, Jeremy S. del Castillo, Aline Xavier da Silveira dos Santos, Qianhui Yu, Paul Noser, Jonas S. Fleck, Bruno Gjeta, Dinko Pavlinic, Simone Picelli, Max Hess, Gregor W. Schmidt, Tom T. A. Lummen, Yanyan Hou, Patricia Galliker, David Goldblum, Marton Balogh, Cameron S. Cowan, Hendrik P. N. Scholl, Botond Roska, Magdalena Renner, Lucas Pelkmans, Barbara Treutlein, and J. Gray Camp. Multimodal spatiotemporal phenotyping of human retinal organoid development. *Nature Biotechnology*, 41(12):1765–1775, 2023. doi: 10.1038/s41587-023-01747-2. URL https://doi.org/10.1038/s41587-023-01747-2.

[42] Daisong Wang, Remi Villenave, Nadine Stokar-Regenscheit, and Hans Clevers. Human organoids as 3D in vitro platforms for drug discovery: Opportunities and challenges. *Nature Reviews Drug Discovery*, 25:204–226, 2026. doi: 10.1038/s41573-025-01317-y. URL https://doi.org/10.1038/s41573-025-01317-y.

[43] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16133–16142, 2023.

[44] Sheng Zhang, Yanbo Xu, Naoto Usuyama, Hanwen Xu, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, Cliff Wong, Andrea Tupini, Yu Wang, Matt Mazzola, Swadheen Shukla, Lars Liden, Jianfeng Gao, Angela Crabtree, Brian Piening, Carlo Bifulco, Matthew P. Lungren, Tristan Naumann, Sheng Wang, and Hoifung Poon. A multimodal biomedical foundation model trained from fifteen million image–text pairs. *NEJM AI*, 2(1):AIoa2400640, 2025. doi: 10.1056/AIoa2400640. URL https://doi.org/10.1056/AIoa2400640.

[45] Zixuan Zhao, Xinyi Chen, Anna M. Dowbaj, Aleksandra Sljukic, Kaitlin Bratlie, Liang Lin, Eliza Li Shan Fong, Gowri Manohari Balachander, Zhaowei Chen, Alice Soragni, Meritxell Huch, Yi Arial Zeng, Qun Wang, and Hanry Yu. Organoids. *Nature Reviews Methods Primers*, 2(1):94, 2022. doi: 10.1038/s43586-022-00174-y. URL `https://doi.org/10.1038/s43586-022-00174-y`.

[46] Eric Zimmermann, Eugene Vorontsov, Julian Viret, Adam Casson, Michal Zelechowski, George Shaikovski, Neil Tenenholtz, James Hall, David Klimstra, Razik Yousfi, et al. Virchow2: Scaling self-supervised mixed magnification models in pathology. *arXiv preprint arXiv:2408.00738*, 2024.

A Technical appendices and supplementary material

A.1 OrgBench

Figure 3 in the main paper visualizes per-task-family relative performance for all 12 backbones; Table 2 below reports the underlying per-(task, dataset) absolute scores. Backbones are abbreviated as DINOv2 (D2), DINOv3 (D3), MAE, Virchow2 (Vchw), CONCH (CN), CellPose-SAM (CP), CLIP (CL), BiomedCLIP (BMC), ResNet-50 (R50), ConvNeXt V2 (CNX), scratch-DINOv2 (sd2), scratch-DINOv3 (sd3). Per-row maxima are in bold; ties at the displayed precision are broken by the unrounded value.

All listed datasets contribute to the per-task-family aggregates of Section 4.1 and Figure 3; two limited datasets are kept in the aggregate but require additional context for interpretation and are flagged here: `organoid_ds_pdac`, where mask evaluation is dataset-limited, and `opara` on T3, which is saturated (every backbone exceeds 0.98 AUC). `organoid_ds_mouse` (n_{train}=10) is not part of the OrgBench evaluation suite and is therefore not in the table; it is used only as a pre-training source (Appendix A.3).

All entries in this table are means over three training seeds as described in Appendix A.4.

Table 2: **OrgBench scores for all 12 backbones.** T1 = mAP@50; T2-inst = mask mAP@50; T2-sem = IoU; T3 = AUC; T4 = weighted F1, where t12–30 is the early window used for the T4 family score and w0–14/74/143 are cumulative windows.

Task	Dataset	D2	D3	MAE	Vchw	CN	CP	CL	BMC	R50	CNX	sd2	sd3
T1 detection (mAP@50)													
	tellu	0.330	**0.700**	0.468	0.294	0.413	0.658	0.265	0.385	0.006	0.114	0.069	0.324
	orgaquant	0.746	0.909	0.902	0.815	0.878	**0.911**	0.807	0.876	0.121	0.604	0.322	0.871
	multiorg	0.559	0.715	0.708	0.652	0.683	**0.735**	0.607	0.664	0.103	0.474	0.291	0.661
T2 instance segmentation (mask mAP@50)													
	orgasegment	0.124	**0.540**	0.313	0.016	0.196	0.532	0.040	0.156	0.000	0.009	0.055	0.434
	orgnet_t	0.011	0.058	0.022	0.003	0.015	**0.068**	0.005	0.013	0.000	0.001	0.002	0.040
	ds_pdac	0.003	0.067	**0.084**	0.000	0.031	0.057	0.001	0.077	0.000	0.000	0.000	0.000
T2 semantic segmentation (IoU)													
	brain1400	0.947	**0.949**	0.937	0.940	0.933	0.947	0.921	0.927	0.840	0.892	0.903	0.875
	cf_resp	**0.585**	0.582	0.375	0.441	0.380	0.554	0.347	0.374	0.135	0.245	0.450	0.451
	orglabel	0.772	0.770	0.658	0.667	0.639	**0.777**	0.603	0.639	0.380	0.459	0.760	0.750
T3 classification (AUC)													
	tellu	0.965	0.948	0.929	0.943	0.937	**0.965**	0.944	0.886	0.917	0.912	0.760	0.819
	opara	1.000	1.000	**1.000**	0.998	1.000	**1.000**	0.999	1.000	1.000	**1.000**	0.984	0.989
	org_clf	**0.866**	0.860	0.838	0.834	0.832	0.842	0.843	0.834	0.813	0.856	0.800	0.811
	afting	0.906	0.977	0.988	0.949	0.885	**0.988**	0.935	0.924	0.913	0.950	0.950	0.948
T4 temporal (weighted F1, afting; per-window mean of the b2_patchstats_transformer per-timepoint output)													
	t12–30	0.454	0.450	0.468	**0.477**	0.454	0.453	0.426	0.306	0.440	0.426	0.368	0.365
	w0–14	0.314	0.313	0.332	0.303	0.312	0.293	0.282	0.261	**0.333**	0.305	0.324	0.297
	w0–74	**0.538**	0.511	0.512	0.532	0.490	0.490	0.501	0.357	0.484	0.484	0.392	0.391
	w0–143	0.652	0.653	**0.655**	0.650	0.586	0.634	0.637	0.505	0.598	0.600	0.530	0.530

Each cell in the T4 rows is the mean weighted F1 over the listed timepoint window, so w0–143 closely matches the overall weighted F1 of Table 9.

Table 3 reports task-family raw scores for the adaptation comparison in Section 4.2, and Table 4 breaks them down by dataset.

Table 3: **Raw scores of base backbones and their adapted versions.** Means over three seeds; each task family is averaged over its datasets (Table 1). T1 = mAP@50; T2-inst = mask mAP@50; T2-sem = IoU; T3 = AUC; T4 = weighted F1 averaged over timepoints 12–30. Norm. is the average normalized score over all 19 configurations (mean ± SD over seeds).

Model	T1	T2-inst	T2-sem	T3	T4	Norm.
DINOv2	0.545	0.046	0.768	0.934	0.454	0.653 ± 0.054
+ adapter (OrgFM-d2)	0.565	0.055	0.775	0.953	0.473	0.741 ± 0.053
+ LoRA	0.559	0.052	0.773	0.949	0.501	0.747 ± 0.012
+ full fine-tuning	0.541	0.054	0.770	0.949	0.442	0.686 ± 0.018
DINOv3	0.775	0.222	0.767	0.946	0.450	0.891 ± 0.042
+ adapter (OrgFM-d3)	0.768	0.244	0.765	0.947	0.498	0.953 ± 0.023
MAE	0.693	0.140	0.656	0.939	0.468	0.725 ± 0.062
+ adapter	0.685	0.143	0.661	0.951	0.456	0.747 ± 0.094
CLIP	0.560	0.015	0.624	0.930	0.426	0.509 ± 0.016
+ adapter	0.620	0.042	0.645	0.925	0.497	0.612 ± 0.021
CellPose-SAM	0.768	0.219	0.759	0.949	0.453	0.892 ± 0.016
+ adapter	0.743	0.212	0.768	0.944	0.421	0.844 ± 0.010

Table 4: **OrgBench scores of the seven adapted configurations.** Mean$_{\pm\mathrm{SD}}$ over the three seeds; T3 is deterministic and shows a single value. Metrics and rows follow Table 2, which lists the base backbones. OrgFM-d2, LoRA, and Full-FT adapt DINOv2, OrgFM-d3 adapts DINOv3, and MAE+ad., CL+ad., and CP+ad. add the same bottleneck adapter to MAE, CLIP, and CellPose-SAM (Appendix A.4).

Task	Dataset	OrgFM-d2	LoRA	Full-FT	OrgFM-d3	MAE+ad.	CL+ad.	CP+ad.
T1 detection (mAP@50)								
	tellu	$0.309_{\pm 0.003}$	$0.300_{\pm 0.006}$	$0.299_{\pm 0.006}$	$0.673_{\pm 0.001}$	$0.454_{\pm 0.005}$	$0.329_{\pm 0.004}$	$0.597_{\pm 0.004}$
	orgaquant	$0.770_{\pm 0.006}$	$0.765_{\pm 0.004}$	$0.745_{\pm 0.004}$	$0.908_{\pm 0.002}$	$0.901_{\pm 0.002}$	$0.865_{\pm 0.000}$	$0.906_{\pm 0.001}$
	multiorg	$0.617_{\pm 0.001}$	$0.611_{\pm 0.002}$	$0.577_{\pm 0.002}$	$0.722_{\pm 0.002}$	$0.700_{\pm 0.009}$	$0.667_{\pm 0.003}$	$0.725_{\pm 0.006}$
T2 instance segmentation (mask mAP@50)								
	orgasegment	$0.151_{\pm 0.006}$	$0.145_{\pm 0.005}$	$0.150_{\pm 0.015}$	$0.556_{\pm 0.006}$	$0.311_{\pm 0.008}$	$0.092_{\pm 0.005}$	$0.470_{\pm 0.003}$
	orgnet_t	$0.008_{\pm 0.004}$	$0.007_{\pm 0.001}$	$0.009_{\pm 0.002}$	$0.061_{\pm 0.001}$	$0.019_{\pm 0.001}$	$0.009_{\pm 0.002}$	$0.064_{\pm 0.001}$
	ds_pdac	$0.007_{\pm 0.009}$	$0.003_{\pm 0.005}$	$0.003_{\pm 0.001}$	$0.115_{\pm 0.029}$	$0.100_{\pm 0.029}$	$0.025_{\pm 0.007}$	$0.102_{\pm 0.015}$
T2 semantic segmentation (IoU)								
	brain1400	$0.948_{\pm 0.000}$	$0.948_{\pm 0.000}$	$0.949_{\pm 0.000}$	$0.947_{\pm 0.000}$	$0.936_{\pm 0.002}$	$0.932_{\pm 0.000}$	$0.946_{\pm 0.001}$
	cf_resp	$0.589_{\pm 0.006}$	$0.588_{\pm 0.002}$	$0.588_{\pm 0.002}$	$0.571_{\pm 0.001}$	$0.391_{\pm 0.002}$	$0.378_{\pm 0.001}$	$0.557_{\pm 0.003}$
	orglabel	$0.788_{\pm 0.003}$	$0.783_{\pm 0.004}$	$0.772_{\pm 0.005}$	$0.777_{\pm 0.001}$	$0.657_{\pm 0.002}$	$0.626_{\pm 0.000}$	$0.799_{\pm 0.002}$
T3 classification (AUC)								
	tellu	0.970	0.962	0.969	0.963	0.958	0.932	0.961
	opara	0.999	0.999	1.000	0.999	1.000	1.000	1.000
	org_clf	0.866	0.855	0.861	0.868	0.854	0.820	0.837
	afting	0.976	0.982	0.967	0.959	0.992	0.949	0.980
T4 temporal (weighted F1, afting; per-window mean of the b2_patchstats_transformer per-timepoint output)								
	t12–30	$0.473_{\pm 0.060}$	$0.501_{\pm 0.018}$	$0.442_{\pm 0.021}$	$0.498_{\pm 0.021}$	$0.456_{\pm 0.093}$	$0.497_{\pm 0.022}$	$0.421_{\pm 0.018}$
	w0–14	$0.357_{\pm 0.028}$	$0.371_{\pm 0.011}$	$0.327_{\pm 0.025}$	$0.384_{\pm 0.025}$	$0.333_{\pm 0.070}$	$0.391_{\pm 0.023}$	$0.289_{\pm 0.013}$
	w0–74	$0.522_{\pm 0.041}$	$0.545_{\pm 0.021}$	$0.499_{\pm 0.018}$	$0.540_{\pm 0.008}$	$0.501_{\pm 0.057}$	$0.541_{\pm 0.015}$	$0.505_{\pm 0.016}$
	w0–143	$0.641_{\pm 0.030}$	$0.651_{\pm 0.004}$	$0.634_{\pm 0.018}$	$0.652_{\pm 0.009}$	$0.643_{\pm 0.034}$	$0.650_{\pm 0.013}$	$0.635_{\pm 0.013}$

Paired differences and subgroups. Table 5 lists, for every cell, the seed-paired difference between each adapted configuration and its frozen base, and averages these differences by task family and by species. For OrgFM-d2 over DINOv2, the mean difference is positive in all five task families and for human, medaka, and mixed-species datasets (the last driven by multiorg), and slightly negative for mouse, driven by tellu T1. Only organoidnetdata (phase contrast) and organolabeler (partly confocal) are not purely bright-field, so we do not form a modality subgroup.

Table 5: **Paired differences between each adapted configuration and its frozen base.** Upper blocks: per cell, the mean over the three seeds of the per-seed difference (adapted minus base). Lower blocks: the per-cell differences averaged within each task family and each species. OrgFM-d2, LoRA, and Full-FT are compared with DINOv2, OrgFM-d3 with DINOv3, and MAE+ad., CL+ad., and CP+ad. with MAE, CLIP, and CellPose-SAM. Species follow Table 1; mixed groups the multi-species datasets multiorg, organoidnetdata, and opara.

Task	Dataset	OrgFM-d2	LoRA	Full-FT	OrgFM-d3	MAE+ad.	CL+ad.	CP+ad.
T1 detection (mAP@50)								
	tellu	−0.021	−0.031	−0.031	−0.027	−0.014	+0.063	−0.061
	orgaquant	+0.024	+0.019	−0.001	−0.001	−0.002	+0.057	−0.005
	multiorg	+0.058	+0.052	+0.018	+0.007	−0.008	+0.059	−0.010
T2 instance segmentation (mask mAP@50)								
	orgasegment	+0.026	+0.021	+0.026	+0.016	−0.002	+0.052	−0.062
	orgnet_t	−0.003	−0.004	−0.001	+0.003	−0.003	+0.004	−0.003
	ds_pdac	+0.003	0.000	−0.001	+0.049	+0.016	+0.024	+0.046
T2 semantic segmentation (IoU)								
	brain1400	+0.001	+0.002	+0.002	−0.001	−0.001	+0.011	0.000
	cf_resp	+0.004	+0.003	+0.004	−0.011	+0.016	+0.031	+0.003
	orglabel	+0.016	+0.012	0.000	+0.007	−0.001	+0.023	+0.022
T3 classification (AUC)								
	tellu	+0.006	−0.003	+0.004	+0.015	+0.029	−0.012	−0.005
	opara	−0.001	−0.001	0.000	−0.001	0.000	+0.001	0.000
	org_clf	−0.001	−0.012	−0.005	+0.008	+0.016	−0.023	−0.005
	afting	+0.070	+0.076	+0.061	−0.018	+0.004	+0.014	−0.007
T4 temporal (weighted F1, afting; per-window mean of the b2_patchstats_transformer per-timepoint output)								
	t12–30	+0.018	+0.047	−0.013	+0.047	−0.012	+0.071	−0.033
Mean over cells, by task family								
	T1	+0.020	+0.014	−0.005	−0.007	−0.008	+0.060	−0.025
	T2-inst	+0.009	+0.006	+0.008	+0.022	+0.004	+0.027	−0.007
	T2-sem	+0.007	+0.006	+0.002	−0.002	+0.005	+0.022	+0.008
	T3	+0.019	+0.015	+0.015	+0.001	+0.012	−0.005	−0.004
	T4	+0.018	+0.047	−0.013	+0.047	−0.012	+0.071	−0.033
Mean over cells, by species								
	human	+0.011	+0.006	+0.004	+0.009	+0.006	+0.025	0.000
	mouse	−0.008	−0.017	−0.013	−0.006	+0.007	+0.026	−0.033
	medaka	+0.044	+0.061	+0.024	+0.015	−0.004	+0.043	−0.020
	mixed	+0.018	+0.016	+0.006	+0.003	−0.003	+0.021	−0.004

A.2 Qualitative examples

Figures 6–11 extend Figure 2. For one dataset per task family, plus the phase-contrast `organoidnetdata` for T2-inst, each row is a randomly chosen test example, the example with the largest paired gain from DINOv2 to OrgFM-d2, the example with the largest paired drop, and, for T3 and T4, the example with the lowest maximum class probability averaged over the two models. The scores are those of Figure 2.

A.3 Dataset construction

Downstream benchmark datasets. Table 1 in the main paper lists all task and dataset configurations. Each dataset is used in its official train/val/test partitions where provided; when a benchmark did not publish splits, we create deterministic splits. For *afting* we construct an experiment-level split that assigns entire experiments (not frames or wells) to train/val/test, preventing implicit leakage from the same biological replicate appearing in multiple partitions.

Per-dataset notes:

opara (§ in Table 1). We adopt publicly available 200-image release (`train-100/` and `test-100/`; $100 + 100 = 200$ images, ten phenotype classes) and re-partition it into 90 train / 10 val / 100 test images.

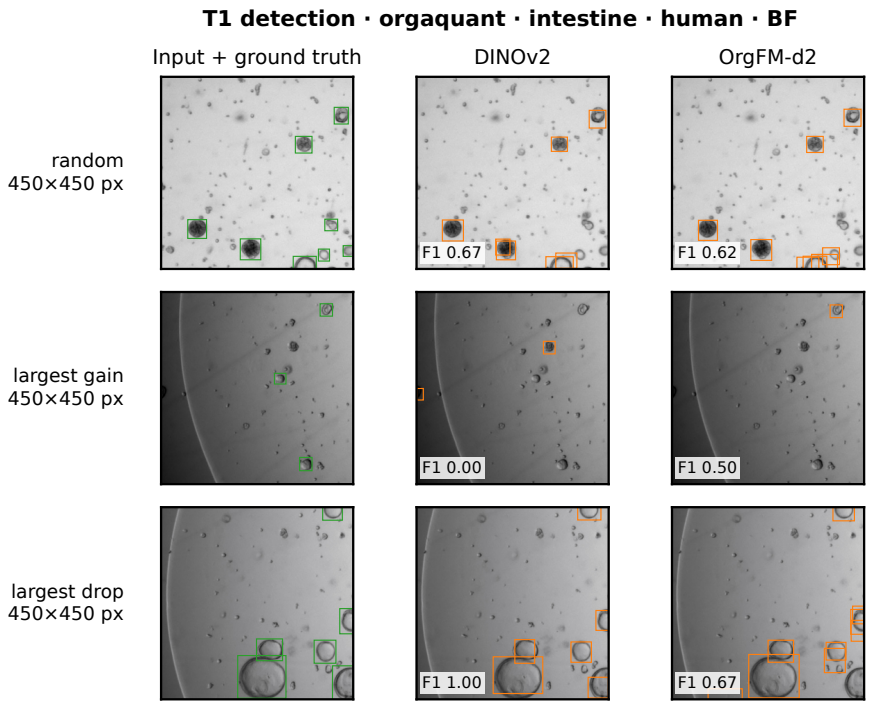

Figure 6: Qualitative T1 detection examples on `orgaquant`: ground truth in green, predictions in orange.

organoid_classifier (§). We adopt publicly available VA22N37 subset (1,500 images, 3 classes), distributed as `train/` (1,200 images) and `test/` (300 images). We use this subset in full and re-partition it into train/val/test.

cf_respiratory. We use `Training/val/` as test (41 images), pool the 786 `Training/train/` images, and hold out 10% as val (78 images), leaving 708 for train.

tellu and *orgasegment.* The per-image counts in Table 1 match the upstream release. The per-annotation counts (23,065 and 15,497 organoids respectively) reflect our processed COCO files, which differ from the upstream totals by at most a few dozen entries after dropping *degenerate annotations* (zero or negative width/height bounding boxes, empty mask polygons, or boxes that fall fully outside the image). These dropped entries cannot be consumed by training and are removed at COCO-build time.

multiorg and *organoidnetdata* are tiled to align effective per-organoid token budget with the ViT-B/14 patch grid; see the dedicated paragraphs below.

Preprocessing pipelines. All images are decoded to 8-bit RGB and resized to each backbone's native input resolution with aspect-ratio preservation, then zero-padded to a multiple of the corresponding patch size. For T1 detection and T2 instance segmentation, the SimpleFPN-style neck derives its multi-scale feature maps from the single backbone output at this resolution; we do not upsample further before the neck. Predicted instance masks are upscaled back to the original image resolution before COCO RLE encoding so that mAP is computed in the same coordinate frame as the ground truth. For T2 semantic segmentation, images are resized with bilinear interpolation and masks with nearest-neighbor. For T3 and T4, inputs are center-cropped/padded to the backbone-native resolution. Bright-field organoid images are kept in their native luminance distribution; no illumination correction is applied.

Illumination-correction sensitivity. Because no illumination correction is applied, we tested whether flat-field correction changes the T3 results (Table 6). The correction is applied after resizing: each RGB channel is divided by a smooth background estimated with a Gaussian filter (σ equal to 1/8

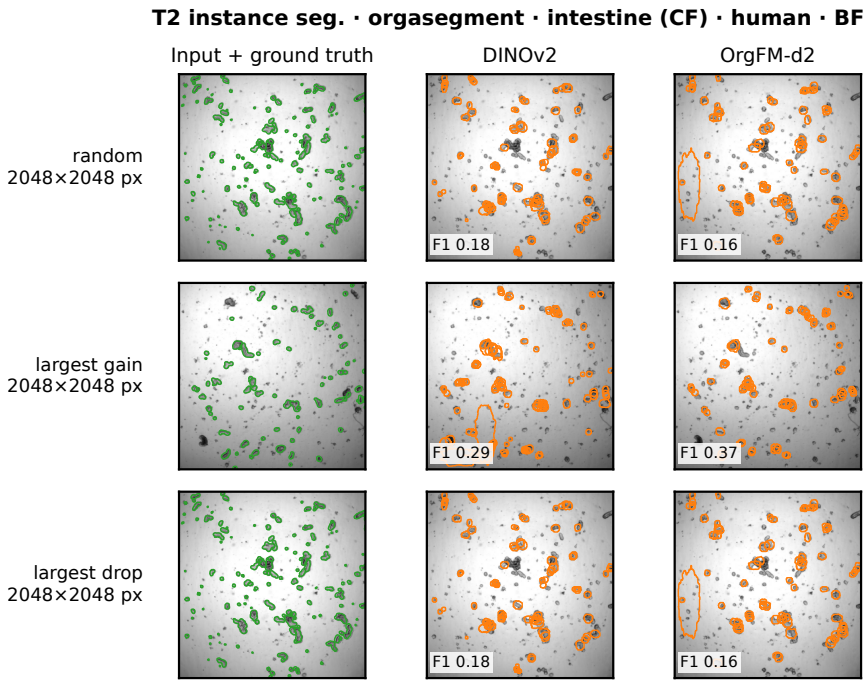

Figure 7: Qualitative T2 instance segmentation examples on `orgasegment`.

of the longer image side), and the result is rescaled to 8 bits between its 1st and 99th percentiles. The parameters are fixed and were not tuned. We re-extract frozen DINOv2 and OrgFM-d2 embeddings from the corrected images and refit the T3 linear probes. Correction changes T3 AUC by at most 0.006 on every dataset. The ordering of DINOv2 and OrgFM-d2 is unchanged on tellu, organoid_classifier, and afting, and on the saturated opara both models stay above 0.999 AUC with and without correction.

Table 6: **T3 sensitivity to flat-field illumination correction.** AUC without (none) and with (flat-field) correction for frozen DINOv2 and OrgFM-d2; the linear probes are refit on the corrected embeddings. Δ is computed before rounding.

	DINOv2			OrgFM-d2		
Dataset	none	flat-field	Δ	none	flat-field	Δ
tellu	0.965	0.967	$+0.002$	0.970	0.972	$+0.002$
opara	1.000	0.999	-0.000	0.999	1.000	$+0.001$
organoid_classifier	0.866	0.869	$+0.003$	0.866	0.867	$+0.001$
afting	0.906	0.912	$+0.006$	0.976	0.971	-0.005

Multi-organ detection dataset (multiorg): tiling. The raw Bukas et al. [4] release provides 411 large bright-field images with $\sim$183 organoids each, a configuration in which a 512-token ViT-B/14 cannot assign a token per organoid; a raw-image control run on this configuration produced near-zero detection mAP for every backbone tested. We therefore preprocess the dataset by cutting each raw image into non-overlapping 512×512 tiles, reprojecting annotations into each tile, and discarding near-uniform background tiles (pixel standard deviation below 10 after 8-bit conversion); tiles without annotations are kept. This yields 36,031 crops (29,886 train / 1,138 val / 5,007 test), which we refer to as "multiorg" throughout the paper. Both the downstream detection head and the SimCLR pre-training pool consume this preprocessed variant; no experiment in the paper uses the raw (un-tiled) release.

PDAC instance-segmentation dataset (organoidnetdata): tiling at evaluation time. The raw Kulkarni et al. [23] release provides phase-contrast images with up to $\sim$70 PDAC organoids each, where individual organoids span only 1–2 patch tokens at the backbones' native input resolution and

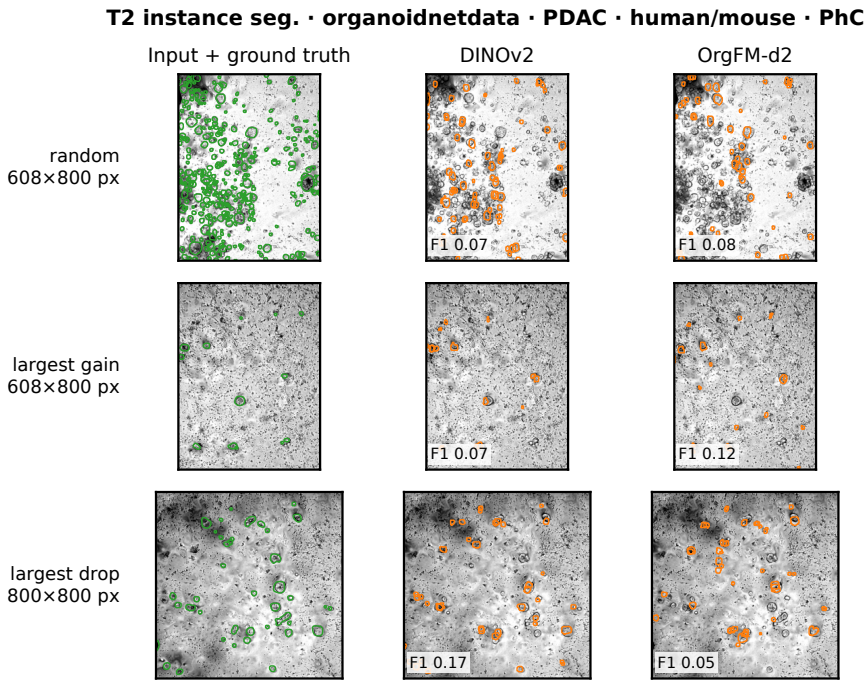

Figure 8: Qualitative T2 instance segmentation examples on the phase-contrast `organoidnetdata` (tiles).

instance segmentation saturates as it does for raw *multiorg*. The release provides its training and validation images as 800×800 patches but its 10 test images at full resolution (1408×1040); we therefore cut each test image into non-overlapping 800×800 tiles (annotations reprojected, edge tiles narrower than half a tile discarded), yielding 912 images (642 train / 250 val / 20 test) with 65,225 instance annotations; this tiled variant is what "organoidnetdata" refers to in all evaluation tables and figures. SimCLR pre-training instead consumes the un-tiled raw images (713 train+val, Table 7). No raw-image instance-segmentation evaluation results are reported in the paper.

Pre-training pool. The SimCLR pre-training pool aggregates 144,339 organoid images drawn from all 13 public datasets, listed as 12 sources in Table 7 because the two OrganoID subsets are pooled, combining each dataset's train and val splits (test splits are excluded). Per-source image counts are re-balanced by $\sqrt{N}$-proportional sampling strategies. Pre-training images are otherwise unmodified beyond basic decoding and resizing depending on the backbone patch size.

Augmentation for pre-training. The two SimCLR views share one augmentation stack and differ only in blur and solarization. Each view is a random resized crop to 224 pixels (scale $[0.32, 1.0]$, bicubic), followed by horizontal and vertical flips and a 90° rotation, each with probability 0.5 (organoid images are orientation-invariant). We then apply one of elastic, grid, or optical distortion (p=0.4), random gamma (p=0.3), Gaussian or ISO noise (p=0.2), CLAHE (p=0.2), brightness and contrast changes of up to 0.15 (p=0.3), defocus or motion blur (p=0.2), coarse dropout of one to five rectangles (p=0.3), color jitter (brightness 0.4, contrast 0.4, saturation 0.05, no hue jitter, since bright-field has narrow color statistics; p=0.8), and grayscale conversion (p=0.2). The second view is additionally blurred with a Gaussian kernel ($\sigma \sim \mathcal{U}[0.1, 2.0]$, p=0.9) and solarized (p=0.2); the first view receives no Gaussian blur.

Augmentation for downstream training. During downstream training, T1 applies horizontal and vertical flips (p=0.5 each) and color jitter (brightness 0.3, contrast 0.3, saturation 0.2, hue 0.02). T3 embeddings are cached from a single forward pass at the backbone's native input resolution (resize plus ImageNet normalization, no flips, crops, or color jitter); the linear probe is fit directly on these

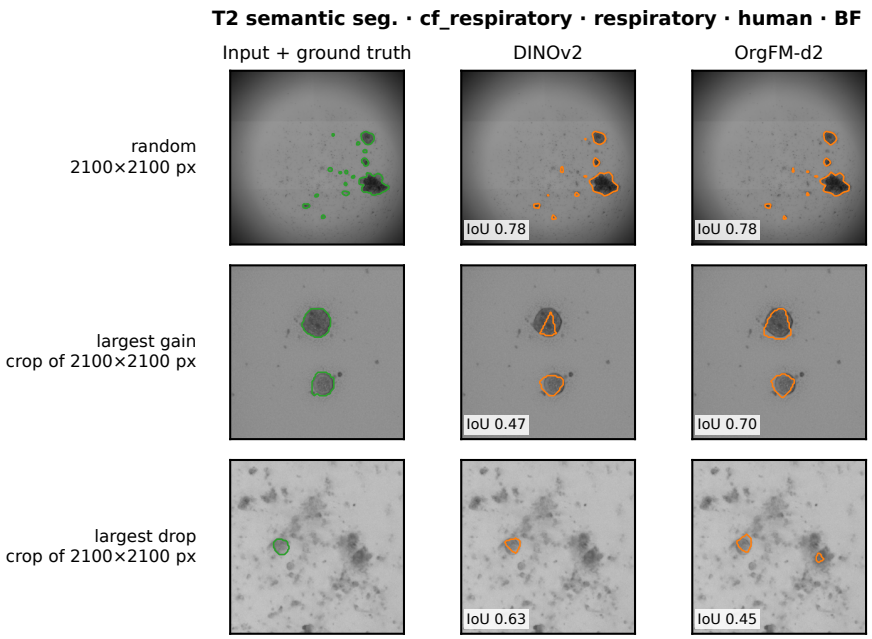

Figure 9: Qualitative T2 semantic segmentation examples on `cf_respiratory`.

Table 7: **Composition of the pre-training pool.** Per-source image counts (train+val) for the pool described in the preceding paragraph. Raw share is each source's fraction of the pool images; $\sqrt{N}$ share is its expected fraction of the training samples under the $\sqrt{N}$-proportional source sampling used for both OrgFM variants.

Source dataset	#Images	Raw share (%)	$\sqrt{N}$ share (%)	Organ coverage
tellu	756	0.5	3.7	intestine (mouse)
orgaquant	1,630	1.1	5.5	intestine (human)
multiorg (tiled)	31,024	21.5	23.8	lung
orgasegment	184	0.1	1.8	intestine / CF
organoidnetdata	713	0.5	3.6	PDAC
organoid_ds (OrganoID)	63	<0.1	1.1	PDAC + mouse intestine
brain1400	1,143	0.8	4.6	brain
cf_respiratory	786	0.5	3.8	respiratory
organolabeler	222	0.2	2.0	brain / enteroid
opara	100	0.1	1.4	multi-organ
organoid_classifier	1,200	0.8	4.7	hypothalamus
afting (time-lapse frames)	106,518	73.8	44.1	retina (medaka)
Total	144,339	100	100	n/a

cached embeddings. T4 operates on cached per-frame embeddings and does not re-augment images. Concept-probe sets are not augmented.

A.4 Training details

Adapter pre-training (OrgFM). Each transformer block of the frozen ViT-B receives two bottleneck adapters (one after the attention sublayer, one after the MLP sublayer), totaling 24 adapter modules across 12 blocks with bottleneck width 2. Per adapter the parameter count is $768\to2$ down-projection plus $2\to768$ up-projection, i.e. $3{,}842$ parameters; the 24-module total is $92{,}208$. A projection head $768\to128\to128$ is trained jointly and discarded after pre-training. Only the adapter and projection head parameters receive gradients. The loss is InfoNCE with $\tau{=}0.10$. We use AdamW with a constant learning rate and weight decay 0.01 (no warmup, no decay; $5{,}000$ samples per

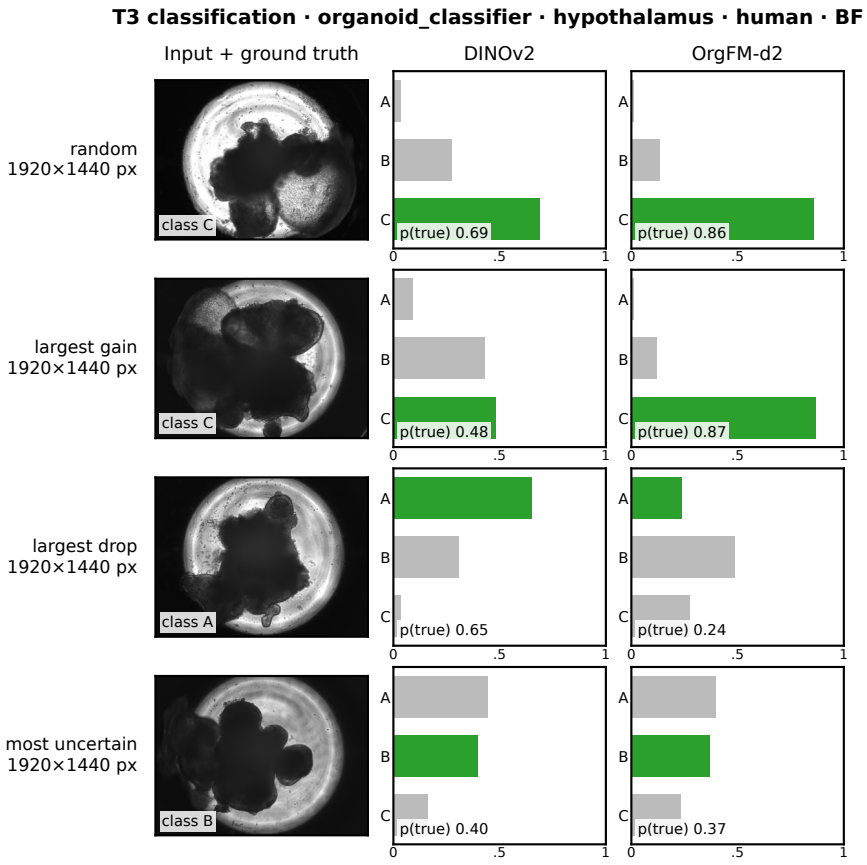

Figure 10: Qualitative T3 classification examples on `organoid_classifier`: predicted class probabilities, with the true class in green.

epoch in both variants). Base learning rates are $1e{-}4$ for (OrgFM-d2) and $2e{-}4$ for (OrgFM-d3), respectively. Batch sizes are 128 (OrgFM-d2) or 512 (OrgFM-d3). Although we train for up to 100 epochs, downstream performance saturates around epoch 25 in both models (the epoch-26 checkpoint has seen $26 \times 5{,}000 = 130{,}000$ samples, roughly one full pass over the pool of $144{,}339$ images). We therefore report results from the epoch-25 checkpoint; later checkpoints are used only for the saturation analysis in Figure 4d.

Adaptation baselines. We compare the adapter with two other ways of adapting the same DINOv2 initialization during continual pre-training: LoRA [19] on the attention `qkv` projection of every block (rank 2, $\alpha{=}2$), and full fine-tuning of all backbone weights. Both reuse the OrgFM-d2 recipe unchanged, such as the sampling strategy and the loss function; only the trainable parameters and the learning rate differ. The learning rate is chosen from $\{1, 2, 5\} \times 10^{-4}$ for LoRA and $\{0.5, 1, 5\} \times 10^{-5}$ for full fine-tuning by the lowest final pre-training loss averaged over three pre-training seeds, without looking at any downstream metric; this selects 5×10^{-4} and 10^{-5}, respectively. Table 8 lists the trainable parameters.

Table 8: **Trainable parameters during continual pre-training of DINOv2 ViT-B/14.** The projection head (114,944 parameters) is trained in every case and discarded afterwards, so it is not counted.

Method	Trainable parameters	At inference
Bottleneck adapter (OrgFM-d2)	92,208	24 adapter modules added
LoRA (`qkv`, rank 2)	73,728	merged into the weights
Full fine-tuning	86,580,480	all weights updated

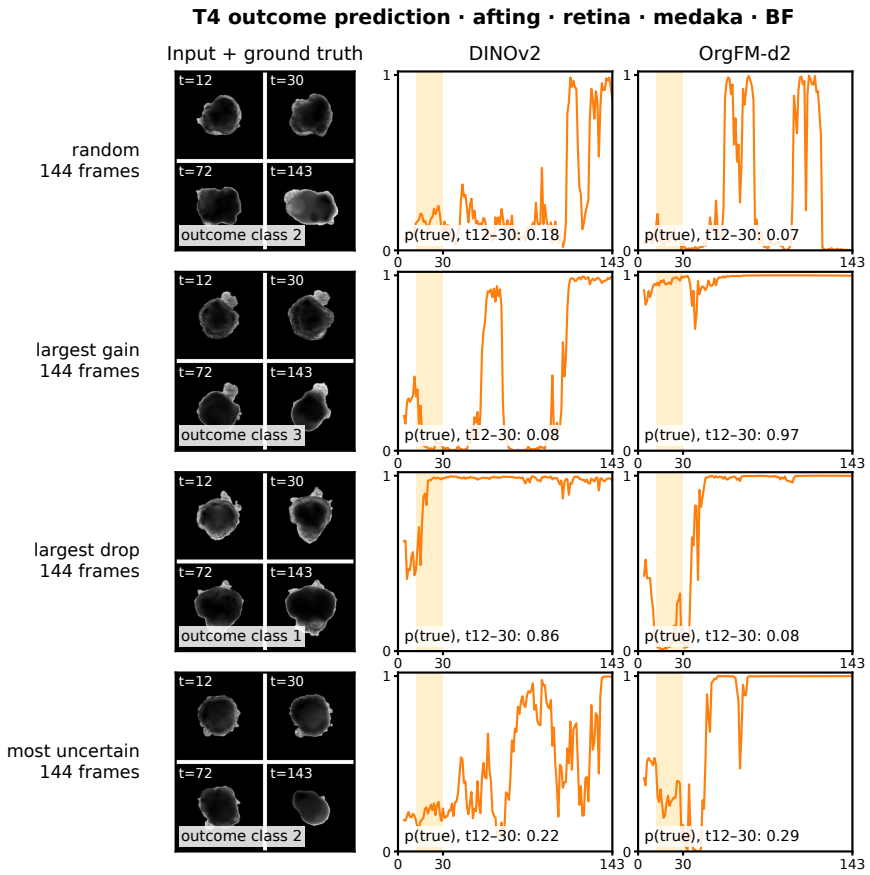

Figure 11: Qualitative T4 outcome prediction examples on `afting`: four frames of the sequence and the predicted probability of the true outcome class over time, with the early window shaded.

Adapter on other backbones. We insert the same bottleneck adapter (width 2, after the attention and MLP sublayers of every block) into frozen MAE ViT-B/16, CLIP ViT-B/16, and CellPose-SAM, which gives 92,208 trainable parameters for MAE and CLIP and 245,856 for the larger CellPose-SAM encoder. Pre-training uses the same pool, augmentation, InfoNCE objective, and number of samples as OrgFM. For MAE and CLIP we search 72 configurations (learning rate $\{0.5, 1, 2\} \times 10^{-4}$, $\tau \in \{0.05, 0.07, 0.10\}$, projector $\{128, 256\}$, batch size $\{128, 512\}$, and equal or $\sqrt{N}$-proportional source sampling). For CellPose-SAM, we set the batch size to 16, and we search learning rate $\{1, 2\} \times 10^{-4}$, $\tau \in \{0.05, 0.10\}$, and projector $\{128, 256\}$ with $\sqrt{N}$ sampling. We select the one with the highest mean T3 validation AUC, without using the test split.

T1 detection. Frozen backbone with a SimpleFPN-style multi-scale neck producing four feature maps at strides $\{p/2,\ p,\ 2p,\ 4p\}$ where p is the backbone patch size, feeding standard Faster R-CNN heads. Input resolution matches the backbone's native input as documented in the preprocessing paragraph above. We use AdamW optimizer with $1\mathrm{e}{-4}$ learning rate, weight decay 0.05, and cosine decay, batch size 4, 50 epochs by default (100 for `multiorg`). Anchors follow the default COCO configuration. mAP@50 is used for evaluation with predicted boxes rescaled to original image resolution. All T1 runs, including every baseline, use the same image augmentation (Appendix A.3).

T2 instance segmentation. Same frozen backbone + ViTDet neck as T1; Mask R-CNN mask head on top of the ROI head. Loss is mask-BCE + box-L1 + cls-CE as standard. Optimizer and schedule match T1, with 100 epochs for `organoidnetdata`. mAP@50 is used for evaluation with predictions rescaled to original resolution before RLE encoding.

T2 semantic segmentation. Frozen ViT features at stride 14 (or 16) are decoded by two 3×3 convolutions (256 channels, GroupNorm, GELU) and a 1×1 convolution to a single foreground logit, which is bilinearly upsampled to the input resolution. The loss is binary cross-entropy plus Dice. AdamW optimizer is used with lr $1e{-4}$, weight decay 0.05, cosine decay, batch size 8, and 50 epochs. IoU is used for evaluation on the test set.

T3 classification. CLS embeddings are standardized on the training split and fit with an ℓ_2-regularized logistic regression whose inverse regularization strength C is chosen from 10 log-spaced values in $[10^{-4}, 10^{4}]$ by 3-fold cross-validation on the training split. Macro one-vs-rest AUC is used for evaluation.

T4 temporal outcome prediction. Per-frame features are cached once from the frozen backbone: for each frame we concatenate the mean, standard deviation, and maximum of the final-layer patch tokens. Each example is a window of five consecutive frames. After a linear input projection, a single-layer pre-norm transformer (width 64, 2 heads, dropout 0.3) with learned temporal position embeddings processes the window; its outputs are mean-pooled over frames and passed to a two-layer MLP that predicts the outcome class. Cross-entropy loss with inverse-frequency class weighting is used. AdamW optimizer is used with lr $5e{-4}$, weight decay 0.05, cosine decay, batch size 64, and 10 epochs. Weighted F1 over outcome classes, per-timepoint or averaged is reported.

Stochasticity and uncertainty. All headline numbers are means over three seeds, paired between each base backbone and its adapted counterpart. Splits are identical across seeds. Note that T3 is a linear probe on frozen features and is deterministic. We report the mean $\pm$ SD over seeds.

A.5 Ensembling adapted backbones (Section 4.2)

We evaluated whether the two adapted backbones (OrgFM-d2 and OrgFM-d3), which share the same SimCLR pre-training pool but start from different base representations, provide complementary outputs that can be combined. Ensembling is performed at the output level. For T1 detection, per-image box predictions from each member are combined with weighted boxes fusion (WBF, IoU threshold 0.55, score threshold 0.01) before COCO evaluation. For T2-inst, per-image mask predictions are concatenated across members and reduced by mask-IoU non-maximum suppression (IoU threshold 0.5) before COCO evaluation. For T2-sem we average per-pixel probabilities, for T3 class probabilities, and for T4 per-class logits. We compare four combinations, named for compactness in Table 9: the *d2-pair* (DINOv2 with its adapted variant OrgFM-d2), the *d3-pair* (DINOv3 with its adapted variant OrgFM-d3), the *OrgFM-pair* (the two adapted variants together), and the *4-model* average (both base backbones plus both adapted variants).

Table 9: **Ensembling base and OrgFM-adapted backbones.** All scores are per-DS averages within each task family; T4 is the weighted F1 over all timepoints. The best single model is underlined and the best ensemble is in bold.

Task	DINOv2	DINOv3	OrgFM-d2	OrgFM-d3	d2-pair	d3-pair	OrgFM-pair	4-model
T1	0.528	<u>0.779</u>	0.564	0.763	0.577	**0.791**	0.728	0.743
T2-inst	0.046	0.230	0.067	<u>0.247</u>	0.057	**0.248**	0.207	0.205
T2-sem	0.768	0.765	<u>0.777</u>	0.764	0.780	0.769	0.782	**0.784**
T3	0.934	0.946	<u>0.953</u>	0.947	0.952	0.952	**0.956**	0.955
T4	0.635	<u>0.660</u>	0.641	0.651	0.652	0.670	0.661	**0.673**

A.6 Ablations

We swept the main SimCLR pre-training hyperparameters and design choices on top of the OrgFM-d2 and OrgFM-d3 backbones, including learning rate ($\{5e{-5}, 1e{-4}, 2e{-4}, 5e{-4}\}$), temperature ($\tau \in \{0.07, 0.10\}$), projector dimension ($\{128, 256\}$), batch size ($\{128, 256, 512\}$), source-sampling strategy (uniform, $\sqrt{N}$-proportional), adapter bottleneck width ($\{2, 4, 8\}$), and training length. Evaluating every configuration on every downstream task (T1–T4) is computationally prohibitive; therefore T1, T2, and T4 were held out from hyperparameter sweep entirely. Across this grid, downstream task-family scores varied by at most $\sim$0.4 pp in T3 mean AUC; therefore the qualitative conclusions are insensitive to the specific grid point.

Leave-one-source-out. For each of the 12 pre-training sources, we repeat the OrgFM-d2 continual pre-training without that source and evaluate T1, T2-sem, T3, and T4 (Figure 4c). Table 10 lists the raw change relative to the full-pool run evaluated with the same pipeline.

Table 10: **Leave-one-source-out changes for OrgFM-d2.** Each entry is the metric with the source removed from the pre-training pool minus the full-pool value in the first row

	T1		T2-sem			T3				T4
Removed source	tellu	orgaq.	brain	cf_resp	orglab	afting	tellu	opara	org_clf	afting
Full pool (value)	0.307	0.769	0.948	0.596	0.788	0.838	0.833	0.950	0.707	0.641
afting	+0.005	−0.004	+0.001	+0.002	−0.001	−0.059	+0.024	+0.010	+0.033	+0.016
brain1400	+0.014	−0.002	0.000	−0.001	+0.001	+0.005	0.000	−0.010	−0.023	+0.016
cf_respiratory	+0.011	+0.001	+0.001	−0.002	0.000	−0.006	+0.024	0.000	−0.013	0.000
multiorg	+0.005	−0.001	0.000	+0.005	−0.007	−0.038	+0.012	−0.020	−0.023	+0.017
opara	+0.005	+0.001	0.000	−0.004	+0.003	−0.017	−0.012	−0.010	0.000	+0.010
organoid_classifier	+0.003	0.000	0.000	+0.005	−0.004	−0.006	−0.024	+0.020	−0.017	+0.023
organoid_ds	+0.011	−0.007	0.000	−0.002	−0.005	−0.017	+0.024	0.000	−0.010	+0.008
organoidnetdata	0.000	−0.001	0.000	+0.001	−0.005	−0.027	−0.024	0.000	−0.020	−0.004
organolabeler	0.000	+0.002	0.000	+0.004	−0.004	+0.026	+0.024	−0.010	−0.003	−0.011
orgaquant	−0.002	+0.001	0.000	+0.003	−0.003	+0.005	−0.036	+0.010	0.000	+0.005
orgasegment	+0.004	−0.001	0.000	−0.001	−0.004	−0.006	−0.036	+0.010	−0.020	+0.019
tellu	+0.002	+0.003	0.000	−0.016	−0.002	−0.017	−0.024	+0.030	−0.017	+0.030

A.7 Interpretation methods

This subsection expands the three interpretation tools introduced in Section 3.3.

Macroscopic probes (Figure 5a-c). Activations are extracted from each of the 12 transformer blocks of the ViT-B backbone on the standard organoid evaluation images (no augmentation), saving the post-block residual stream prior to the next attention layer.

- *Layer-wise CKA* between base and adapted backbones is computed on a fixed set of N=2,048 patch tokens drawn uniformly from the held-out organoid pre-training validation pool, using linear CKA with centered Gram matrices.
- *Layer-wise linear probing* fits a logistic regression with L2 regularization (C=1.0) on the layer-ℓ CLS token; we report mean AUC over 3 seeds with the train/val/test split documented in Section A.3, evaluated on each of the 4 T3 datasets and aggregated.
- *Concept probes* fit ridge regressions (5-fold CV on $\alpha \in \{10^{-2}, \ldots, 10^{2}\}$) from frozen patch tokens to four scalar morphology primitives (size in pixels, mean brightness, eccentricity, circularity) computed per image from an Otsu-thresholded foreground mask of the input image (T3 datasets are classification-only and do not provide segmentation ground truth). We report best-layer R^2 on a held-out test split, averaged over all T3 datasets.

Layer knockout (Figure 5d). For each transformer block $\ell \in \{0, \ldots, 11\}$, we zero the up-projection weights of both adapter modules inserted in block ℓ (the post-attention adapter and the post-MLP adapter), leaving all adapters in the other 11 blocks intact. Because each bottleneck adapter is a residual addition, zeroing its up-projection makes the adapter exactly equivalent to skipping it. We then evaluate T3 (logistic-regression head) without retraining the head. An *all-12* knockout (every adapter zeroed) establishes the upper bound on the cumulative adaptation effect.

Crosscoder training (Figure 5e-h). Crosscoders are trained per layer at L3, L7, and L11 (the input to block 3, block 7, and the output of the last block, respectively) on 5,000 images sampled from the same organoid pre-training pool (excluded test images; `afting` frames, stored in HDF5, were not sampled), with input pairs $(h_\ell^{\text{base}}, h_\ell^{\text{orgfm}})$ formed from base and adapted ViT *patch-token* activations on the same images (the CLS token is excluded from the decomposition). The Crosscoder is a top-k sparse autoencoder with a single shared encoder and two decoders, $W_{\text{dec}}^{\text{base}}$, $W_{\text{dec}}^{\text{orgfm}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{dict}}}$, one per side. Features are classified using the ratio between decoder column norms, $r_i = \|W_{\text{dec}}^{\text{orgfm}}[:$

$, i]\|/\|W_{\text{dec}}^{\text{base}}[:, i]\|$, restricted to features whose mean activation frequency over the training pool is at least 10^{-5} (alive features): *OrgFM-specific* ($r_i \geq 2$), *base-specific* ($r_i \leq 0.5$), and *shared* ($0.5 < r_i < 2$). Training settings are as follows: dictionary size $8\times$ (d_{dict}=6144 at d_{model}=768) with k=128 used uniformly across the qualitative split (Figure 5e, f) and the knockout study (Figure 5g, h); Adam optimizer, lr 3×10^{-4}, batch 4,096 patch tokens, 50,000 steps (about 30 passes over the sampled patches). Two validation controls accompany every run: (i) a *self–self* control (both inputs come from the same model) drives OrgFM-specific and base-specific feature counts to $\leq 1\%$ of the dictionary, confirming the decomposition is identifying genuine cross-model signal; (ii) a $4\times/8\times$ dictionary sensitivity sweep yields specific-feature ratios within ± 1 pp.

Per-feature task attribution (Figure 5f). For each Crosscoder feature i, we ask how strongly the trained T3 head reads out its OrgFM-side decoder direction. The Crosscoder is trained on patch-token activations, and its decoder columns $\{d_i\}$ are vectors in the same d_{model}=768 residual space as the CLS token at this layer, so we use them as residual-stream directions and ask whether the T3 head, which reads CLS, places weight along them.

Composing the trained head weights $W_{\text{task}} \in \mathbb{R}^{C \times d_{\text{model}}}$ with the OrgFM-side decoder $W_{\text{dec}}^{\text{orgfm}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{dict}}}$ gives

$$A \;=\; W_{\text{task}}\, W_{\text{dec}}^{\text{orgfm}} \;\in\; \mathbb{R}^{C \times d_{\text{dict}}}, \qquad A_{c,i} = w_c^{\top} d_i.$$

Concretely, shifting the L11 CLS embedding from h_{CLS} to $h_{\text{CLS}} + \alpha\, d_i$ changes the class-c logit by $\alpha A_{c,i}$, so $A_{c,i}$ is the decoder-scaled sensitivity of the class-c head to direction d_i — a static quantity computed from the fitted head and the decoder column, with no forward pass.

We summarize each feature i by its mean absolute sensitivity across classes, $a_i = \frac{1}{C}\sum_c |A_{c,i}|$, and within each Crosscoder category by $\bar{a}_{\text{cat}} = \text{mean}_{i \in \text{cat}}\, a_i$. Figure 5f reports the ratio $\bar{a}_{\text{OrgFM-specific}}/\bar{a}_{\text{shared}}$ on the three non-saturated T3 datasets (`afting`, `tellu`, `organoid_classifier`); we omit `opara`, where T3 AUC saturates above 0.997 and head-weight magnitudes become hard to interpret.

Feature knockout (Figure 5g, h). We use the $8\times/k$=128 Crosscoder at L11. The two panels use complementary intervention protocols, dictated by which representation the downstream head consumes.

Figure 5g: the T3 head is a linear classifier on the CLS embedding, and L11 is the final block, so any modification to patch tokens at L11 has no remaining attention layer through which to reach the CLS. Standard activation patching is therefore not applicable here, and we instead intervene directly on the cached CLS embedding. The intervention exploits the fact that, although the Crosscoder is trained on patch tokens, its decoder columns are 768-dimensional vectors in the same residual coordinate system that hosts the CLS token at L11; we therefore use the OrgFM-side decoder columns of the OrgFM-specific features as candidate direction vectors and project them out of the CLS embedding: $\tilde{h}_{\text{cls}} = (I - QQ^{\top})h_{\text{cls}}$, where Q is an orthonormal basis (via QR decomposition) of these decoder columns. This is not a claim that the Crosscoder reconstructs the CLS embedding; it is a directional probe asking whether the patch-token-derived OrgFM-specific subspace is also a subspace the CLS-based T3 head reads from. We then refit the logistic-regression head on the projected training embeddings and evaluate on the projected test embeddings; if these directions carry information used by T3, the refit accuracy should drop. As a control, we project out the same number of randomly chosen alive *shared* directions and average over 10 random subsets, which isolates the effect of removing OrgFM-specific information from the generic effect of reducing the embedding's effective dimensionality.

Figure 5h: at L11 we replace the OrgFM patch-token activations with the Crosscoder reconstruction in which the OrgFM-specific feature columns are zeroed, recompute the patch-statistics representation (mean, std, max over patch tokens) used by the temporal transformer, and feed it through the pre-trained T4 transformer *without retraining*. The random-knockout control zeros the same number of randomly chosen alive features (mean over 5 subsets); a *recon control* re-injects the un-zeroed Crosscoder reconstruction to isolate reconstruction error from the knockout effect.

Visualizing Crosscoder feature dictionaries. To make the three feature categories from Section 3.3 concrete, Figure 12 visualizes top activations for randomly drawn *shared*, *base-specific*, and *OrgFM-*

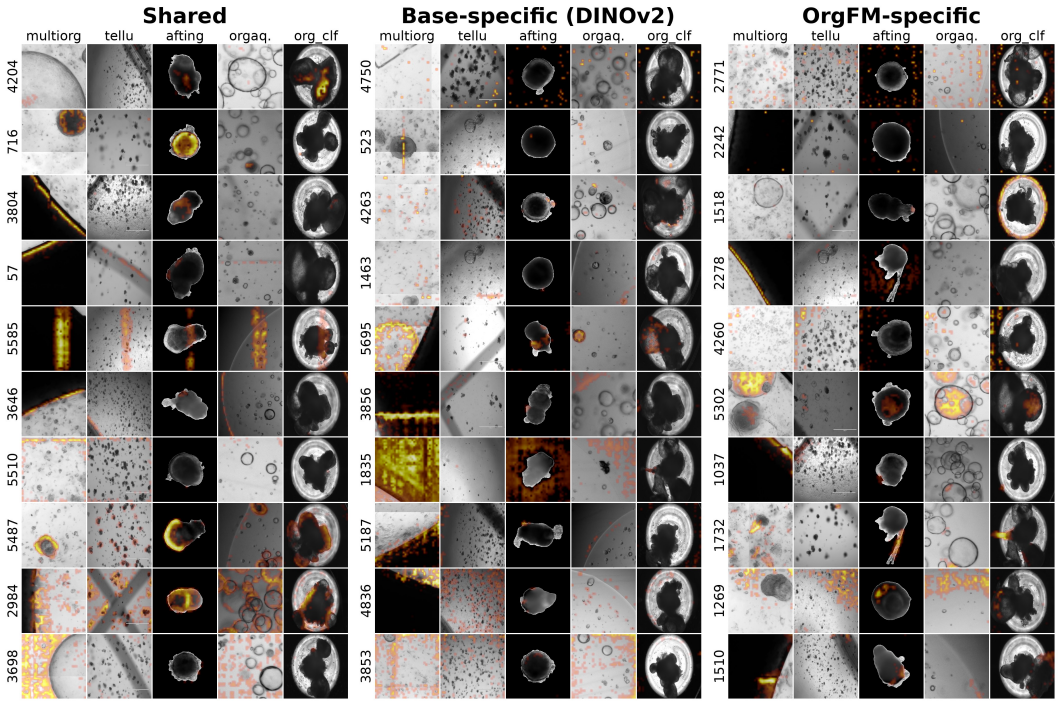

Figure 12: **Crosscoder feature visualizations at L11 ($8\times$/k=128, the same dictionary used for all panels of Figure 5e–h).** Each row shows one feature, labeled by its index, and each tile the image with the highest activation of that feature in one dataset (orgaq. = orgaquant; org_clf = organoid_classifier), with the patch-level activation heatmap overlaid on one color scale per feature; blocks are categories (left: *shared*; middle: *base-specific (DINOv2)*; right: *OrgFM-specific*). For each category we display ten features drawn at random with a fixed seed from those active in all five datasets, in decreasing order of decoder norm. Some features respond to organoid interiors or boundaries and others to non-organoid structure such as well rims, image seams, or background.

specific features at L11 on five datasets; the selection of features and images is given in the figure caption.

A.8 Compute

All training and analysis runs use a single NVIDIA H200 (143 GB) GPU per job. The OrgFM SimCLR adapter pre-training (144,339 images, the epoch-25 checkpoint used throughout the paper) takes $\sim$4–5 h on $1\times$ H200. All downstream-task and interpretation runs (T1–T4 heads, Crosscoders, knockouts) range roughly from seconds (T3 linear probes on cached embeddings) to single-GPU-day scale (multi-epoch dense-prediction heads on the larger detection / instance-segmentation datasets). Total wall-clock to reproduce every number in the paper is on the order of a few thousand H200-hours.

A.9 Limitations

Our evaluation focuses on the specific organoid-imaging design space: 2D imagery, and few adapter families trained with one self-supervised objective. Whether the same recipe transfers to larger backbones, to alternative adapter designs or objectives, and to volumetric imagery are open questions. We also fixed the self-supervised objective to SimCLR to isolate the effect of the adaptation parameterization, and do not claim that SimCLR is optimal. T4 temporal outcome prediction is also evaluated on a single dataset, so the early-window result should be reproduced on additional temporal organoid datasets. Our interpretation analysis covers only the DINOv2/OrgFM-d2 pair on part of tasks. Absolute instance-segmentation performance also remains low, so the reported gains indicate more transferable representations rather than readiness for fully automated morphometric analysis.

The benchmark inherits biases from its source datasets: annotations come from different laboratories under different protocols, and we do not re-measure inter-annotator agreement, so some performance ceilings may reflect label noise rather than backbone limits. The pre-training pool is also imbalanced across organs and species, reflecting what is publicly available rather than a deliberate balance. Every source except organoidnetdata (phase contrast) and part of organolabeler (confocal) is bright-field. Technical imaging conditions, such as the microscope, illumination, and signal-to-noise ratio, also differ across datasets and are confounded with organ, species, and laboratory, so differences in performance across datasets cannot be attributed to biology alone.

A.10 Broader impact

OrgBench and OrgFM lower the engineering barrier to high-quality organoid image analysis: the frozen-backbone-plus-adapter recipe is trainable on a single GPU, making the workflow accessible to laboratories without dedicated deep-learning resources. Better organoid assays accelerate drug screening, disease modeling, and regenerative-medicine research, and improving in-vitro readouts can in turn reduce reliance on animal experiments. The benchmark, adapter checkpoints, splits, and code are released openly to support reproducibility.

The main risks are around mis-use and over-claiming. Miscalibrated model predictions could mislead biological or pre-clinical decisions if used without external validation, and OrgFM is explicitly not a medical device: it has not been audited for clinical use and should not directly inform patient care. Outputs should be treated as hypotheses to be confirmed by domain experts and downstream assays, and the model card distributed with the checkpoints disclaims clinical and regulatory use.

The pre-training pool and benchmark coverage inherit the biases of publicly available organoid datasets, which over-represent certain organs (intestine, retina, brain), species (medaka, mouse, a small set of human cell lines), and laboratory protocols. Adapters trained on this pool may transfer less well to under-represented settings, and benchmark conclusions about a “best” backbone are conditional on this distribution rather than universal; we encourage users working in new organ or species settings to verify performance on a held-out internal split before relying on the released checkpoints.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state three contributions matching our experiments: (i) OrgBench; (ii) an adapter (OrgFM) trained by SimCLR continual pre-training that improves its base backbone on the cross-task aggregate; and (iii) a mechanistic analysis of the DINOv2/OrgFM-d2 pair on T3 and T4 using targeted feature and layer knockouts.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix A.9 lists: (i) the adapter study focuses on ViT-B backbones, 2D imagery, and one self-supervised objective; (ii) T4 temporal prediction is evaluated on a single dataset (afting, 988 sequences); (iii) the pre-training pool is imbalanced and imaging conditions are confounded with biology; (iv) the interpretation analysis covers only the DINOv2/OrgFM-d2 pair on T3 and T4.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper is purely empirical (benchmarking, continual pre-training, and mechanistic analysis). No theorems or formal proofs are introduced.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3 (OrgBench construction and evaluation protocol in Section 3.1) documents the full pipeline.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code for the full training and evaluation pipeline, with a README, is released at `https://github.com/yu-takagi/OrganoidFM`. All benchmark datasets are pre-existing public releases.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3.1 and Appendix A.4 report optimizer, learning rates, batch sizes, per-dataset epoch budgets, input resolution, augmentation, data splits, and the seed protocol. Pre-training details are documented in Section 3.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All headline numbers are means over three downstream-training seeds, paired between each base backbone and its adapted counterpart, and are reported with the standard deviation over seeds; base-versus-adapted comparisons also report seed-paired differences (Appendix A.1, Table 5). The seed varies head initialization, data order, dropout, and image augmentation on fixed splits. T3 is a deterministic linear probe.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A.8 reports GPU type and memory and wall-clock time per run.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses only publicly released organoid imaging datasets under their original licenses; no new human-subject data was collected. We reviewed the NeurIPS Code of Ethics and confirm conformance with respect to dataset usage and privacy.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Appendix A.10 discusses positive impacts (lowering the barrier to high-quality organoid image analysis, accelerating drug screening and developmental biology research) and potential negative impacts (miscalibrated model predictions influencing downstream biological or pre-clinical decisions if used without validation). We position OrgFM as a research tool and note that it is not validated for clinical or diagnostic use.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The released assets are a benchmark protocol and an adapter for non-clinical bright-field organoid microscopy. The adapter is a discriminative feature extractor (not generative) built on publicly released backbones, trained on publicly released microscopy data, and intended for research use. We therefore consider the misuse risk to be low and do not apply access gating.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and backbones used in the paper, each with its original citation are listed in main text and appendix. All assets are publicly released.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release (i) the evaluation protocol, per-task metric implementations, and split files and (ii) adapter checkpoints.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve any crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No new human-subject research was conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as an important, original, or non-standard component of the core methodology. The benchmark, the adapter, and mechanistic analysis do not depend on LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.