Who Needs Labels? Adapting Vision Foundation Models With the Metadata You Already Have

Elouan Gardès[1,2] **Seung Eun Yi**[1] **Kartik Ahuja**[1]

Théo Moutakanni[1] **Huy V. Vo**[1] **Piotr Bojanowski**[1] **Wolfgang M. Pernice**[3]*

Loïc Landrieu[2]* **Camille Couprie**[1]*

[1]Meta FAIR [2]LIGM, CNRS, Univ Gustave Eiffel, ENPC, IPP [3]Columbia University, New York *Equal supervision

Abstract

We propose a label-free approach to adapt powerful but generic vision foundation models to specialized scientific domains. Standard supervised fine-tuning is often ill-suited to these settings: labels are scarce, and task-specific training can reduce the generality needed to cope with complex domain shifts. We instead leverage metadata to adapt representations to new domains in a weakly-supervised manner. Our method, FINO, combines a standard self-supervised objective with flexible metadata guidance that handles both highly granular discrete metadata and continuous metadata. It encourages the representation to preserve informative factors while suppressing spurious ones. Across subcellular fluorescence microscopy, Earth observation, wildlife monitoring, and medical imaging, FINO outperforms matched self-supervised and fully supervised adaptation baselines. Complete-adaptation repeats confirm consistent performance across domains. Without task labels for backbone adaptation, FINO also obtains higher reported scores than selected specialised systems on three domains and matches the fourth under differing cross-paper protocols.
Code is available at `github.com/facebookresearch/dinov3/tree/FINO`.

1 Introduction

Visual foundation models provide rich, general representations that perform well across many natural-image tasks [1, 2, 3]. However, they often degrade on specialised domains such as biomedical microscopy [4, 5] and Earth observation [6], whose distributions differ substantially from natural images [7]. These datasets also exhibit subtle but pervasive domain shifts [8] that arise from both meaningful variation, such as geography [9] or staining protocols [10], and spurious factors such as sensor characteristics or reagent aliquot, often called *batch effects* in biology [11]. Fortunately, such datasets are also typically accompanied by rich metadata correlated to these variations, and which are available at virtually no extra cost.

A standard way to adapt foundation models to a new domain is supervised fine-tuning [12, 13, 14, 15]. In scientific settings, however, labels are often scarce or expensive [16], and fine-tuning can overfit task-specific signals, hurting out-of-distribution (OOD) generalisation [17]. In the process, it can erase much of the generality of the pretrained model, occasionally yielding worse performance than zero-shot inference. Unsupervised domain adaptation instead leverages unlabelled data from the target distribution [18, 19], but remains task-driven and can likewise induce task-specific collapse [20, 21].

In this work, we study *representation adaptation*: specialising a pretrained foundation model to a new application domain without task labels, then evaluating the frozen adapted representation with lightweight probes (Fig. 1). Purely self-supervised adaptation can struggle under distribution shift, as

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

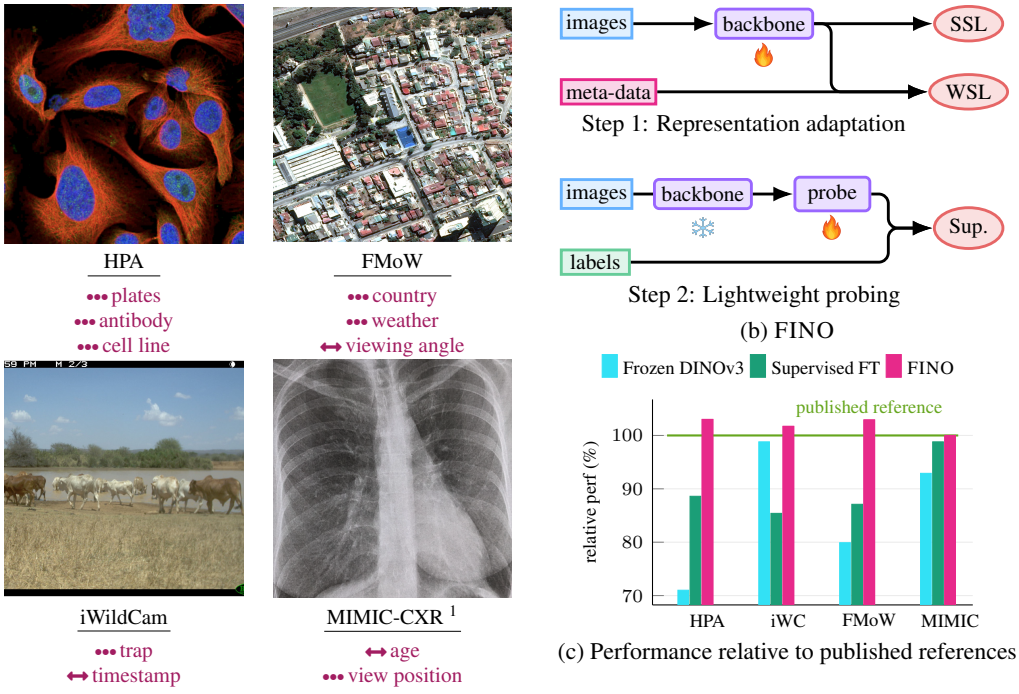


Figure 1: **Learning with metadata.** Scientific datasets (a) often include metadata describing their content and acquisition. FINO uses these signals as weak supervision (WSL), together with a self-supervised objective (SSL), to adapt a generic foundation model without task labels (b). In matched evaluations, its frozen representations outperform supervised fine-tuning; the right panel compares FINO means to reported specialised-system results under differing protocols (c).

its representations remain sensitive to low-level or context-specific cues [5, 22]. Recent work has shown that metadata can help steer self-supervised adaptation toward more robust representations [5, 22, 23, 24, 25]. However, existing approaches remain partial: they are often tailored to a single modality or assume only a small number of domains, whereas scientific metadata is highly granular, spanning thousands of discrete categories or continuous values.

We propose FINO (FIne tuning with NO labels), a unified framework for metadata-driven representation adaptation (Fig. 1). It augments a self-supervised objective with metadata-guided losses and targeted regularisation to preserve informative structure while suppressing spurious variation. By combining prototype-based contrastive guidance, self-supervision, and appropriate regularisation, our method scales to heterogeneous and highly granular metadata, where classical domain generalisation methods struggle.

Empirically, FINO improves OOD performance over matched self-supervised, supervised, and domain-adaptation baselines without using unlabelled test data. Complete-adaptation repeats confirm consistent performance across domains, and metadata guidance outperforms task-label guidance in the tested FMoW and HPA settings. FINO also obtains higher reported scores than selected specialised systems on HPA [26], FMoW [27], and iWildCam [28], and matches the MIMIC-CXR [29] reference under differing data, supervision, tuning, and inference protocols. The objective and loss weights are shared across domains, with dataset-specific metadata and evaluation and a documented iWildCam schedule exception. Our contributions are:

- FINO, a unified metadata-guided adaptation framework for discrete and continuous metadata that requires neither task labels nor metadata at inference time; and
- controlled evidence across four scientific domains that metadata guidance improves task-agnostic representations over matched adaptation baselines, supported by complete-adaptation repeats, metadata-assignment evaluations, and task-label-supervised comparisons.

[1] The X-ray illustration is not from MIMIC-CXR, but a personal Xray from one of the authors.

2 Related work

Self-supervised Representation Learning. Self-supervised learning (SSL) has enabled the emergence of powerful visual representations through contrastive [30, 31], self-distillation [32, 1], bootstrapping [33], masking [34], and joint-embedding predictive approaches [35]. Several methods incorporate prototype-based objectives to structure representations, either in a fully unsupervised manner [36, 37] or with supervision [38]. A common remedy to mitigate degradation when transferring to specialised domains such as microscopy [39, 40] or remote sensing [41] is supervised fine-tuning [12, 13, 14, 15], but it relies on task-specific annotations that are often scarce in scientific settings and may lead to overfitting to spurious correlations [42, 43]. This motivates adapting representations using weak and freely available signals such as metadata.

Domain Generalisation and Invariant Representations. Batch effects (systematic variations induced by experimental conditions) are a well-documented source of domain shift in scientific data [44], and are known to significantly impact learned representations, including in self-supervised settings [40, 5]. A large body of work, therefore, seeks to learn representations invariant to predefined domain labels by explicitly removing domain-specific information. In supervised settings, this is typically achieved through feature alignment across domains [45], gradient reversal mechanisms [18], contrastive fairness objectives [46], or compositional attribute decompositions [47]. In the self-supervised setting, Scalbert et al. [48] propose to standardise Fourier statistics across batches. Complementary to these approaches, post-hoc methods such as Harmony [49] and Symphony [50] correct batch effects in latent space.

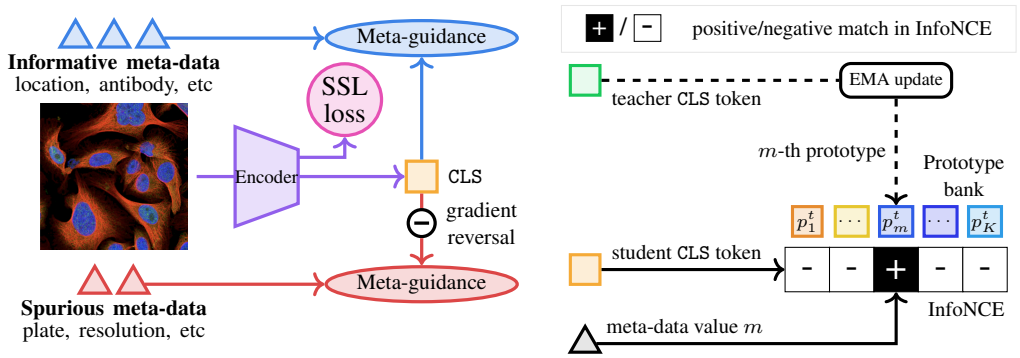
These approaches treat each selected factor as a nuisance to be removed, and do not provide a mechanism to preserve or encourage informative metadata within the same framework. They also typically assume a small number of discrete domains, whereas scientific metadata often spans thousands of classes or continuous values. In contrast, we propose a unified metadata-guided framework that encourages informative factors while suppressing spurious ones, using prototype-based guidance [38] for high-cardinality discrete metadata and learned regressors for continuous factors.

Metadata Guidance in Scientific Imaging. Metadata can be viewed as a form of weak supervision for representation learning in domain-specific settings [51]. In microscopy and related medical imaging, such signals have been exploited through protein identity [52], antibody labels [53], microscopy-specific consistency objectives [22], and patient metadata in retinal imaging [54, 24]. These methods show the promise of metadata, but remain tied to specific modalities, task formulations, or supervision regimes. In remote sensing, SatMAE [55] and Scale-MAE [56] incorporate temporal, spectral, or resolution metadata through modified positional encodings. However, these methods require metadata at test time. Closer to our approach, SatMIPS [23] proposes a metadata-guided CLIP-like framework [2] for satellite imagery, but is specifically tailored to continuous geospatial metadata. Side information extends beyond image-attached acquisition fields. Prior work uses patient identity [57], paired radiology reports [58], biological taxonomy [59], image geolocation [60], naturally co-occurring modalities [61], and wildlife observations as ecological context for satellite imagery [62]. These signals differ in availability and deployment requirements: FINO focuses on structured metadata available during adaptation and not required at inference. Framing each field as an auxiliary objective is also related to multi-task learning [63]; unlike contextual-token methods [64], our framework does not modify the backbone, and unlike language-guided sampling [65], it directly accommodates both discrete and continuous factors at scale.

3 Method

Our goal is to adapt the representation of a pretrained foundation model to a target application domain without task labels and independently of any downstream task. To this end, we leverage freely available metadata to encourage the model to organise its features along informative factors while reducing sensitivity to spurious ones.

Setting. Our approach is based on the standard DINO self-supervised setup[1], with a student–teacher architecture operating on augmented views and a teacher updated by exponential moving average. As shown in Fig. 2, we extend this framework to accommodate metadata supervision. Let



(a) Metadata-Driven Representation Adaptation (b) Discrete metadata guidance for metadata type t

Figure 2: **FINO.** Left: overall representation adaptation pipeline, which combines DINO, iBOT, metadata guidance (informative factors $\mathbb{M}_+$ encouraged, spurious factors $\mathbb{M}_-$ suppressed via gradient reversal), and SIGReg to adapt a pretrained encoder without task labels. Right: discrete metadata guidance for one metadata type t. The student embedding is contrasted against an EMA prototype bank indexed by the metadata value, and the matched prototype is updated from the teacher embedding. Continuous metadata uses a predictor head instead of a prototype bank.

x be an image and ϕ a backbone encoder. We denote by $\phi(x) \in \mathbb{R}^D$ the student embedding (*e.g.*, the class token of a ViT), and by $\phi_{\text{teacher}}(x)$ the corresponding teacher embedding. Each sample is associated with one or several metadata types $t \in \mathcal{T}$, which may be discrete (*e.g.*, antibody, country) or continuous (*e.g.*, timestamp, viewing angle).

Informative and spurious metadata. We partition metadata types into informative factors $\mathbb{M}_+$, which the representation should preserve, and spurious factors $\mathbb{M}_-$, which it should suppress. Informative metadata capture meaningful structure in the data distribution and correlates with variables of interest. For example, in fluorescence microscopy, the antibody or staining protocol determines which cellular structures are highlighted and is therefore predictive of protein localisation. In Earth observation, geographic location influences both label distributions (land use, crop types) and their appearance. Organising representations along such factors provides a meaningful supervisory signal. In contrast, spurious metadata correspond to variations introduced by the acquisition process that are not related to image content. For instance, plate identifiers in high-throughput microscopy often encode batch effects (*e.g.*, illumination or preparation artifacts), while sensor characteristics or viewing conditions in satellite imagery reflect acquisition settings rather than scene content. Representations that encode such factors tend to generalise poorly beyond conditions observed in training [5].

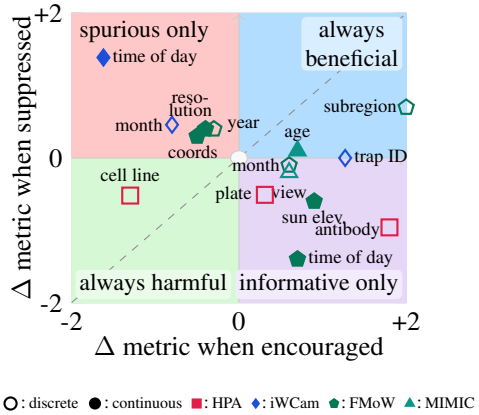
Choosing metadata factors. Assigning metadata to $\mathbb{M}_+$ or $\mathbb{M}_-$ is application-dependent but does not require target-domain labels. We use prior knowledge, desired deployment invariances, and held-out-validation sweeps when needed; no test metric is used for metadata, checkpoint, or hyperparameter selection. The final configurations encourage antibody on HPA, encourage subregion and suppress year on FMoW, suppress time of day on iWildCam, and encourage view position on MIMIC-CXR. Figure 3 evaluates all 15 candidate factors separately under both signs. Most have a clear beneficial sign; HPA cell line is harmful under either sign because it is entangled with antibody identity (App. A.4.2). Before suppressing correlated fields, we audit their source-only dependence, group redundant fields, and retain one representative at the desired granularity (App. A.4.3).

Metadata guidance. We define a guidance loss $\mathcal{L}^{(t)}_{\text{meta}}$ for each metadata type t. Its form depends on whether the metadata is discrete or continuous, and its sign determines whether it encourages alignment with informative factors or suppresses spurious ones via reversed gradients.

Discrete metadata. For a discrete metadata type t taking values in $\{1, \dots, M\}$, we maintain a prototype bank $\{p^t_m\}_{m=1}^M$, where each prototype corresponds to one possible metadata value. For an image x with value m for metadata t, we define the following prototype-based contrastive objective, inspired by supervised contrastive learning [38] and prototypical contrastive learning [37]:

$$\mathcal{L}^{(t)}_{\text{meta}}(x) = -\log \frac{\exp\left(\langle \phi(x), p^t_m \rangle / \tau\right)}{\sum_{n=1}^{M} \exp\left(\langle \phi(x), p^t_n \rangle / \tau\right)}, \tag{1}$$

Figure 3: **Impact of Metadata Guidance.** We measure the impact of adding individual metadata factor when encouraged (x-axis, assigned to $\mathbb{M}_+$) or suppressed (y-axis, assigned to $\mathbb{M}_-$), reported as the Δ on each dataset's target metric relative to a no-guidance baseline. The quadrants characterise each factor's role: *always beneficial* factors help in both regimes; *informative only* and *spurious only* factors help only under correct assignment; *always harmful* factors degrade performance regardless of assignment, occupied only by cell line on HPA, an entanglement artefact analysed in A.4.2. The dashed diagonal separates factors that benefit more from encouragement (below) vs suppression (above). See A.4 for a detailed per-factor analysis.



where $\tau > 0$ is a temperature parameter, $\phi(x)$ and p_m^t are ℓ_2-normalised so that $\langle \cdot, \cdot \rangle$ is a cosine similarity (see App. A.5). Unlike standard supervised contrastive learning, no positive or negative samples are needed within the batch, allowing it to scale to metadata with large numbers of classes M. Prototypes are updated via an exponential moving average of teacher embeddings:

$$p_m^t \leftarrow \alpha p_m^t + (1-\alpha)\ \phi_{\text{teacher}}(x)\ , \tag{2}$$

where α is the EMA update rate; p_m^t is detached and re-projected onto the unit sphere after each update. The approach is illustrated in Fig. 2.

Continuous metadata. For a metadata t with continuous values $m \in \mathbb{R}^d$, we learn a predictor $g^{(t)}$ (*e.g.*, an MLP) and define a regression loss:

$$\mathcal{L}_{\text{meta}}^{(t)}(x) = \left\| g^{(t)}\left(\phi(x)\right) - m \right\|_2^2 \ . \tag{3}$$

Encouraging and suppressing metadata. We use the same metadata loss for all metadata types, but control its effect on the encoder via gradient reversal. For informative metadata $t \in \mathbb{M}_+$, the encoder minimises $\mathcal{L}_{\text{meta}}^{(t)}$, encouraging alignment with the corresponding factor. For spurious metadata $t \in \mathbb{M}_-$, we reverse the gradient before it reaches the encoder, effectively discouraging the representation from encoding this information. When the guidance branch includes trainable parameters (*e.g.*, the predictor $g^{(t)}$), they are always optimised to minimise $\mathcal{L}_{\text{meta}}^{(t)}$, following the standard adversarial formulation of Ganin and Lempitsky [18].

Balancing metadata losses. Discrete and continuous metadata losses have different scales and gradient profiles, which can hinder joint training. When multiple metadata branches are used jointly, we therefore balance their gradients with a simple EMA-smoothed multiplicative factor, in the spirit of prior gradient-balancing methods [66, 67, 68]. Full details are provided in App. A.5.

Overall objective. We build on a standard self-supervised setup combining the DINO and iBOT losses respectively on the CLS and patch tokens. Following LeJEPA [69], we additionally regularise the pre-normalisation DINO bottleneck with SIGReg (see App. A.5). The training objective is

$$\mathcal{L}(x) = \mathcal{L}_{\text{DINO}}(x) + \mathcal{L}_{\text{iBOT}}(x) + \lambda_{\text{SIGReg}}\mathcal{L}_{\text{SIGReg}}(x) + \sum_{t \in \mathbb{M}_+ \cup \mathbb{M}_-} \lambda_{\text{meta}}^{(t)} \mathcal{L}_{\text{meta}}^{(t)}(x)\ , \tag{4}$$

where $\{\lambda_{\text{meta}}^{(t)}\}_t$, and λ_{SIGReg} are nonnegative weights. In practice these weights are not tuned: λ_{SIGReg} is taken as-is from the LeJEPA paper [69], and a single value $\lambda_{\text{meta}}^{(t)}$=0.03 is reused for every metadata branch across all our application domains (see App. A.5), meaning the method can be applied to a new dataset without an extra hyperparameter sweep. For $t \in \mathbb{M}_-$, gradients from $\mathcal{L}_{\text{meta}}^{(t)}$ are reversed before reaching the encoder. See Fig. A.2 for a pseudo-code implementation.

Implementation details. We initialise the backbone from an open-source DINOv3 [70] ViT-L checkpoint obtained by distilling a larger 7B-parameter teacher. As this checkpoint provides only backbone

weights, we initialise all heads (DINO, iBOT, and metadata) from random weights.Adaptation is one unlabeled job with two regimes. First, the randomly initialised DINO, iBOT, and metadata heads and the patch embedding are trained while the transformer blocks remain frozen. The backbone is then unfrozen with a linear learning-rate warm-up and the full model is trained end-to-end. The frozen downstream probe is evaluation rather than a representation-training phase. This schedule prevents random heads from destabilising the pretrained backbone; App. A.5 gives the full protocol and its iWildCam exception.

4 Results

4.1 Datasets and Evaluation

Datasets. We evaluate our approach on four datasets from distinct scientific domains: fluorescence microscopy with Human Protein Atlas (HPA) [71, 72, 26], Earth observation with Functional Map of the World (FMoW) [27, 17], wildlife monitoring with iWildCam [28], and chest X-ray analysis with MIMIC-CXR [29]. Additionally, we evaluate cross-dataset transfer within the same application domain with three additional datasets: OpenCell [73], CheXpert [74], and FLAIR-Hub [75]. Detailed descriptions of all datasets, splits, metadata, and evaluation protocols are provided in App. A.3.

Evaluation Protocol. We use the same ViT-L backbone (306M parameters) for all methods. Unless specified otherwise, models are initialised from a public DINOv3 checkpoint [70]; see Appendix App. A.2 for a study of generalisation to other backbones. We compare our approach to two classes of methods: task-centric domain adaptation and representation adaptation. For task-centric methods, the model is trained using labelled data from the training set, optionally combined with robustness-inducing regularisation losses. For unsupervised domain adaptation (UDA), we additionally leverage unlabelled data from the test set. In both cases, we follow the standard protocol of fully fine-tuning all model parameters; note that this involves learning a task-specific prediction head. In contrast, in *representation adaptation* we aim to improve the intrinsic quality of learned representations independent of a specific target task or dataset. We first adapt model weights using unlabelled source data and, optionally, metadata. The model is then frozen, and a lightweight attentive [76] or linear probe is trained on top. The linear probe concatenates the CLS token with the average-pooled patch embeddings and applies a single linear layer. The attentive probe uses a single cross-attention layer that attends to patch embeddings from the last four transformer blocks and aggregates them into a single representation. Both probes remain lightweight (2.7M parameters for the attentive probe; under 1M for the linear probe) compared to a full fine-tuning of the model.

4.2 Results and Analysis

We compare our approach to existing adaptation methods across several application domains, analyse the role of different metadata factors and their effect on the learned representation space, and further evaluate the method in cross-dataset evaluation and low-supervision settings.

Adaptation performance. We report the performance of all adaptation methods in Tab. 1. First, training a ViT from scratch is generally not viable on the considered datasets. Second, despite the domain gap between its pretraining data and our target domains, fine-tuning a pretrained DINOv3 backbone yields inconsistent gains over probing the frozen model, and even degrades performance on iWildCam. Unsupervised domain adaptation methods yield inconsistent gains over supervised fine-tuning: substantially improving on FMoW but underperforming on iWildCam, with broadly comparable performance on HPA and MIMIC, with the exception of CORAL on HPA. Applying the standard DINO self-supervised adaptation recipe generally outperforms supervised fine-tuning, but its gains remain inconsistent, even underperforming the frozen DINO baseline on iWildCam.

In contrast, FINO is the strongest evaluated method under the matched backbone, data, and evaluation protocol. The complete-run means are 52.9 ± 0.8 on iWildCam, 53.3 ± 0.8 on FMoW, and 81.7 ± 0.1 on MIMIC; the full-resolution HPA result is 61.2. All configurations, metadata assignments (Tab. 2), checkpoints, and probe hyperparameters were selected on held-out validation data; neither HPA Kaggle public nor private scores informed selection. Against the listed published references, FINO reports higher scores on HPA, iWildCam, and FMoW and matches MIMIC under differing upstream data, supervision, resolution, search, and inference protocols. The objective and loss weights

Table 1: **Quantitative results.** Matched adaptations and published references: Ouyang et al. [26] for HPA, Choi et al. [42] for iWildCam/FMoW, and Moutakanni et al. [77] for MIMIC-CXR. Published references report mean±standard deviation where available. FINO reports mean±standard deviation over 8, 5, and 5 complete runs on iWildCam, FMoW, and MIMIC. *HPA is the full-high-resolution run; App. A.1 reports its low-resolution repeat. "Param." counts parameters trained with labels; ❄/🔥 denote frozen/trainable backbones. WGA is worst-group accuracy.

Type	Method	Label supervision		HPA	iWildCam	FMoW	MIMIC
		Param.	Backbone	*F1*	*OOD F1*	*WGA*	*AUROC*
Published reference		80–300M	🔥/❄	59.4	52.0±0.4	51.8±0.4	81.7±0.1
Supervised training	From scratch	306M	🔥	44.7	10.1	24.3	76.8
	Fine-tuning	306M	🔥	52.6	43.4	40.5	80.7
Domain adapt. and general.	DANN	306M	🔥	50.2	38.5	46.2	—
	CORAL	306M	🔥	41.1	38.5	45.9	—
	Group-DRO	306M	🔥	52.6	26.1	46.4	80.0
Representation adaptation	Frozen DINOv3	2.7M	❄	42.2	51.4	45.0	75.9
	DINO (SSL)	2.7M	❄	60.0	50.0	48.4	81.1
	FINO (ours)	2.7M	❄	**61.2***	**52.9±0.8**	**53.3±0.8**	**81.7±0.1**

Table 2: **Final metadata assignments and selection signals.** A dash denotes no factor in that branch. Signs, checkpoints, and probe hyperparameters were chosen with the listed held-out metric; no test-set labels or test metrics were used for selection.

Dataset	Informative $\mathbb{M}_+$	Spurious $\mathbb{M}_-$	Held-out selection metric
HPA	antibody	—	5-fold cell-line-split macro-F1
iWildCam	—	time of day	OOD-validation macro-F1
FMoW	subregion	year	OOD-validation WGA
MIMIC-CXR	view position	—	IID-validation AUROC

Table 3: **All-positive metadata guidance without sign selection.** Every available metadata field is encouraged, with no informative/spurious selection. HPA uses the low-resolution configuration.

Dataset	Metric	Factors	No guide	Curated	All positive
FMoW	OOD WGA	13	48.4	**53.3**	48.4
HPA (224^2)	private F1	3	53.0	**54.7**	**54.7**
iWildCam	OOD macro-F1	4	50.0	**52.9**	52.7

are shared across domains, with dataset-specific metadata/evaluation and the iWildCam schedule exception in App. A.5.

Which metadata helps? We report in Fig. 3 the impact of each metadata factor when treated as either informative or spurious. Some metadata are consistently informative: using them as supervisory signals improves performance, indicating that they capture meaningful structure aligned with downstream tasks. In contrast, some metadata are purely spurious: performance improves only when their influence is suppressed, suggesting that they encode nuisance variations that hinder generalisation. Interestingly, certain metadata are beneficial in both regimes: both predicting and actively suppressing these factors provide useful training signals, acting as auxiliary objectives that steer the self-supervised loss toward more semantic representations. A few metadata factors can be harmful in both regimes because they entangle task-relevant and acquisition structure; App. A.4 gives the full factor-level analysis.

Is metadata selection necessary? As a no-selection stress test, Tab. 3 encourages every available metadata field. All-positive guidance matches or exceeds unguided SSL on all three tested datasets. It matches the curated configuration on low-resolution HPA and is within 0.2 points on iWildCam; on FMoW, curation improves WGA by 4.9 points when the 13 available factors include overlapping geographic variables. All-positive guidance is therefore a robust no-selection baseline, while curation pays off for highly redundant metadata.

Table 4: **Metadata versus task-label guidance.** Same adaptation and evaluation; only the guidance signal changes. Differences are relative to SSL adaptation without guidance. Metadata alone is best on both datasets, and adding task labels to metadata hurts. HPA uses the low-resolution configuration; full IID/OOD results and the multi-label HPA setup are in App. A.1.

Guidance during adaptation	FMoW OOD WGA	HPA OOD F1
None (SSL adaptation)	48.4	53.0
SupCon (task labels)	48.3 ↓−0.1	54.5 ↑+1.5
Task-label prototypes	48.5 ↑+0.1	53.1 ↑+0.1
SupCon + selected metadata	51.0 ↑+2.6	54.0 ↑+1
FINO (metadata only)	**53.3** ↑+4.9	**54.7** ↑+1.7

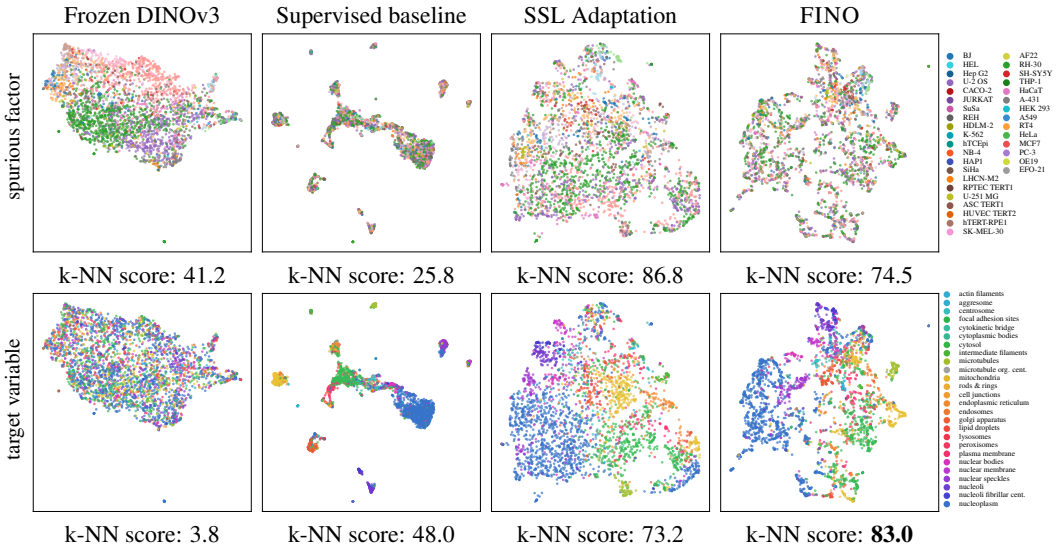


Figure 4: **HPA representation geometry.** UMAPs of 3,000 validation images with a single protein-location label, coloured by cell line (top) or protein localisation (bottom). The displayed k-NN scores use k=100 for cell line and k=10 for localisation. FINO, guided only by antibody identity, strengthens target structure while attenuating cell-line structure without directly suppressing cell line.

Task-label guidance. If downstream task labels are available, should they guide adaptation instead of metadata? Since they define the evaluated task, one would expect them to guide better. We test this alongside our DINO and iBOT objectives, keeping all other evaluation protocols fixed. We compare metadata-only FINO with conventional SupCon [38], task labels used through FINO's prototype objective, and SupCon combined with the selected metadata. For multi-label HPA, SupCon positives share at least one active label, while prototype guidance updates every active class prototype. We find that metadata guidance not only competes with task-label guidance, but actually outperforms it on both FMoW and HPA. Adding SupCon to metadata also does not improve over metadata guidance alone. This suggests that the metadata framework provides a real robustness benefit over using downstream labels, even when those labels are available during adaptation.

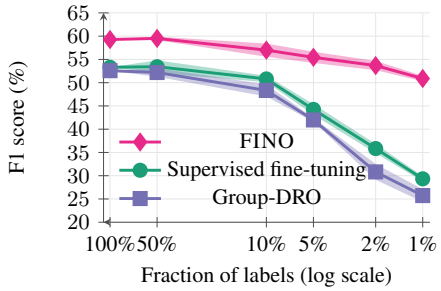
Representation geometry. Fig. 4 visualises the HPA feature space. The off-the-shelf DINOv3 representation is poorly organised by protein localisation (4% k-NN). Fully supervised fine-tuning produces label-aligned clusters but generalises poorly to the OOD test set (48%). Self-supervised adaptation remains strongly structured by cell line (87%). Antibody-guided FINO raises protein-localisation k-NN accuracy to 83% and reduces cell-line accuracy to 75%, despite never using protein-location labels during adaptation or explicitly suppressing cell line.

Low-supervision regime. As shown in Fig. 5, FINO consistently outperforms all baselines across all label fractions, with particularly large gains in the low-data regime. When only 1–10% of labels are available, performance remains high (50.9–57.0%), while supervised fine-tuning and Group DRO degrade sharply (*e.g.*, 53.3% → 29.3% and 52.6% → 25.7% at 1%). Notably, FINO also surpasses fully supervised fine-tuning even at 100% of labels, indicating that label-free representation

Table 5: **Transfer learning.** A DINOv3 model is adapted on HPA, FMoW, or MIMIC-CXR and then frozen, probed, and evaluated on OpenCell, FLAIR-Hub, or CheXpert, respectively.

Method	HPA → OpenCell [73] 100×ARI	FMoW → FLAIR-Hub [75] mIoU	MIMIC-CXR → CheXpert [74] AUROC
Frozen DINOv3	46.6	62.0	87.0
Supervised adaptation	53.0	61.9	87.3
FINO	**58.8**	**62.3**	**88.0**

Figure 5: **Low-supervision regime.** F1 score on HPA as a function of the fraction of labelled data (out of 94,270 annotated images) used to train the downstream predictor. We compare FINO, supervised fine-tuning, and Group DRO (using cell lines as groups). Results use a single fold without cross-validation, so absolute scores differ from Tab. 1. Each point is averaged over 5 seeds (stratified label samplings), except at 100% where the full label set is deterministic. Shaded areas indicate the standard deviation across these seeds.



adaptation can produce features that are both more robust and better aligned with the task. In contrast, Group DRO provides limited gains and remains below standard supervised training, suggesting that robustness to predefined groups is insufficient in this setting.

Transfer learning. A key advantage of label-free adaptation is that the representation does not collapse around a fixed label set. This improves robustness to domain shift and enables transfer across datasets within the same application domain but with different class definitions. In Tab. 5, we adapt a generic DINOv3 model using data and metadata from one dataset, then probe and evaluate on a different dataset from the same domain (Fig. A.1 summarises the structural differences between each transfer pair). When input channels differ between datasets, we drop unsupported input channels or their corresponding model weights, *e.g.*, the nIR band in FLAIR-Hub or the microtubule and ER channels absent from OpenCell. For OpenCell, we report the Adjusted Rand Index (ARI) between protein-level embeddings and ground-truth localisation clusters, following the protocol of Kobayashi et al. [52]. We observe that supervised adaptation is either stable or improves over a frozen DINOv3 backbone, while FINO consistently improves over both. On OpenCell, the off-the-shelf DINOv3 backbone alone already exceeds prior published methods; FINO adds a further $+0.122$ ARI on top, despite never being trained on OpenCell data (App. A.9).

4.3 Ablation Study

We evaluate in Tab. 6 the impact of the main components of FINO on the HPA benchmark, and in Tab. 7 its dependence on the backbone.

- *Loss terms.* Removing metadata guidance reduces F1 by 1.7, while replacing SIGReg with KoLeo reduces it by 0.9. Removing both produces a larger 2.8-point decrease, showing that metadata guidance and feature regularisation make complementary contributions.
- *Pretraining.* Initialising from DINOv3 rather than training the backbone from scratch improves F1 by 3.3, showing that general-purpose visual features remain valuable even after substantial adaptation to fluorescence microscopy.
- *Prototype guidance.* Replacing the EMA prototype bank with an MLP classifier reduces F1 by 3.3. The momentum-updated prototypes therefore provide a stronger mechanism for learning from high-cardinality discrete metadata than direct metadata classification.
- *Backbone.* Replacing DINOv3 with an image–text-pretrained SigLIP2 ViT-L/16, with all hyperparameters unchanged, preserves the same ordering on all four datasets (Tab. 7): SSL adaptation improves over the frozen backbone, and metadata guidance adds a further $+1.6$ to $+4.5$ points. The benefit of FINO is therefore not tied to the DINOv3 recipe (App. A.2).

Table 6: **Ablation on HPA Kaggle private set.** The full FINO row is the mean±standard deviation over 10 complete low-resolution runs; ablation rows are single runs and deltas are relative to that mean. All rows use one fold at 224^2, rather than the 768^2 full-resolution HPA setting in Tab. 1.

meta-data	regularisation	pre-trained	discrete guide	F1-score	
✓	SIGReg	✓	proto	**54.7±0.7**	(ours)
✗	SIGReg	✓	N/A	53.0	↓−1.7
✓	KoLeo	✓	proto	53.8	↓−0.9
✗	KoLeo	✓	N/A	51.9	↓−2.8
✓	SIGReg	✗	proto	51.4	↓−3.3
✓	SIGReg	✓	MLP	51.4	↓−3.3

Table 7: **SigLIP2 backbone.** Same pipeline and hyperparameters as with DINOv3, primary metric per dataset as in Tab. 1.

SigLIP2 backbone	HPA F1	iWildCam OOD F1	FMoW WGA	MIMIC AUROC
Frozen	38.0	11.3	30.0	71.5
SSL adaptation ($\mathcal{L}_{\text{DINO}}$)	45.7	16.9	35.1	77.6
FINO	**47.3**	**21.4**	**36.8**	**79.7**

- *Compute.* The metadata heads add a negligible overhead: on a single H100 GPU with batch size 64, FINO runs at 0.76 s/iter, against 0.74 s/iter for pure SSL adaptation (App. A.6).

Together, these ablations show that FINO's performance does not come from a single component: metadata guidance, regularisation, pretrained visual features, and prototype-based supervision each contribute to the final representation.

5 Conclusion

We introduced *representation adaptation* and FINO, which uses freely available metadata to specialise foundation models without task labels. Across four scientific domains, FINO improves over matched supervised, domain-adaptation, and self-supervised baselines, with complete-adaptation repeats demonstrating stable performance. On FMoW and HPA, metadata provides stronger OOD guidance than conventional SupCon, task-label prototypes, or their combination with metadata. FINO also remains effective with limited labels and transfers across datasets and prediction tasks, showing that its representations are reusable rather than tied to one label space. Together, these results establish metadata as practical supervision for robust adaptation, while highlighting the need to distinguish informative, spurious, redundant, and entangled factors.

Broader impact. Because FINO uses metadata as weak supervision, it may reproduce or amplify biases encoded by sensitive or imbalanced attributes, such as gender or geographic origin. Treating such factors as informative can lead to uneven performance across groups. Metadata should therefore be selected carefully and audited beyond aggregate performance, particularly in medical and geographic applications. All datasets used in this work are established research benchmarks, and we do not release new personal data.

Reproducibility. Our public repository, `https://github.com/facebookresearch/dinov3/tree/FINO`, provides the end-to-end code for data and metadata download, preprocessing, dataloading, every training stage, probing, and evaluation, together with exact configurations, environments, splits, seeds, commands, metrics, and table-generation scripts. The appendix documents the corresponding protocols. Laboratory policy prevents redistribution of trained checkpoints.

Acknowledgments and Disclosure of Funding

This work was partially supported by NIH/NHGRI award 5R00HG011488-05 to WMAP, Hi!PARIS (ANR-23-IACL-0005), and the PEPR SHARP (PEPR IA, ANR-23-PEIA-0008).

References

[1] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024.

[2] 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 *ICML*, 2021.

[3] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv:2502.14786*, 2025.

[4] 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), 2025.

[5] Wolfgang M Pernice, Michael Doron, Alex Quach, Aditya Pratapa, Sultan Kenjeyev, Nicholas De Veaux, Michio Hirano, and Juan C Caicedo. Out of distribution generalization via interventional style transfer in single-cell microscopy. In *CVPR Workshop on Computer Vision for Microscopy Image Analysis (CVMI)*, 2023.

[6] Kumar Ayush, Burak Uzkent, Chenlin Meng, Kumar Tanmay, Marshall Burke, David Lobell, and Stefano Ermon. Geography-aware self-supervised learning. In *ICCV*, 2021.

[7] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*. IEEE, 2011.

[8] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, 2022.

[9] Abhishek Kuriyal, Elliot Vincent, Mathieu Aubry, and Loic Landrieu. CoDEx: Combining domain expertise for spatial generalization in satellite image analysis. In *CVPR workshop EarthVision*, 2025.

[10] Mehrdad Noori, Gustavo A. Vargas Hakim, David Osowiechi, Fereshteh Shakeri, Ali Bahri, Moslem Yazdanpanah, Sahar Dastani, Ismail Ben Ayed, and Christian Desrosiers. Histopath-C: Towards realistic domain shifts for histopathology vision-language adaptation. In *WACV*, 2026.

[11] Karin Stacke, Gabriel Eilertsen, Jonas Unger, and Claes Lundström. Measuring domain shift for deep learning in histopathology. *IEEE Journal of Biomedical and Health Informatics*, 25(2): 325–336, 2021.

[12] Benjamin P Veasey and Amir A Amini. Parameter-efficient fine-tuning of DINOv2 vision transformers for lung nodule classification. In *International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2024.

[13] Andreas Peter Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? data, augmentation, and regularization in vision transformers. *TMLR*, 2022.

[14] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. In *NeurIPS*, 2019.

[15] Yanyan Huang, Weiqin Zhao, Zhengyu Zhang, Yihang Chen, Yu Fu, Feng Wu, Yuming Jiang, Li Liang, Shujun Wang, and Lequan Yu. Knowledge-guided adaptation of pathology foundation models effectively improves cross-domain generalization and demographic fairness. *Nature Communications*, 2025.

[16] Devin P Sullivan, Casper F Winsnes, Lovisa Åkesson, Martin Hjelmare, Mikaela Wiking, Rutger Schutten, Linzi Campbell, Hjalti Leifsson, Scott Rhodes, Andie Nordgren, et al. Deep learning is combined with massive-scale citizen science to improve large-scale image classification. *Nature biotechnology*, 36(9):820–828, 2018.

[17] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

[18] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

[19] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*. PMLR, 2015.

[20] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *IJCNN*, pages 1–8. IEEE, 2020.

[21] Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *ICLR*, 2021.

[22] Johan Fredin Haslum, Christos Matsoukas, Karl-Johan Leuchowius, Erik Müllers, and Kevin Smith. Metadata-guided consistency learning for high content images. In *Medical Imaging with Deep Learning*, pages 918–936. PMLR, 2024.

[23] Jules Bourcier, Gohar Dashyan, Karteek Alahari, and Jocelyn Chanussot. Learning representations of satellite images from metadata supervision. In *ECCV*. Springer, 2024.

[24] Robbie Holland, Oliver Leingang, Hrvoje Bogunović, Sophie Riedl, Lars Fritsche, Toby Prevost, Hendrik P N Scholl, Ursula Schmidt-Erfurth, Sobha Sivaprasad, Andrew J Lotery, Daniel Rueckert, and Martin J Menten. Metadata-enhanced contrastive learning from retinal optical coherence tomography images. *Medical Image Analysis*, 2024.

[25] Haiyan Lan, Qiaoxi Zhu, Jian Guan, Yuming Wei, and Wenwu Wang. Hierarchical metadata information constrained self-supervised learning for anomalous sound detection under domain shift. In *ICASSP*, 2024.

[26] Wei Ouyang, Casper F Winsnes, Martin Hjelmare, Anthony J Cesnik, Lovisa Åkesson, Hao Xu, Devin P Sullivan, Shubin Dai, Jun Lan, Park Jinmo, Shaikat M Galib, Christof Henkel, Kevin Hwang, Dmytro Poplavskiy, Bojan Tunguz, Russell D Wolfinger, Yinzheng Gu, Chuanpeng Li, Jinbin Xie, Dmitry Buslov, Sergei Fironov, Alexander Kiselev, Dmytro Panchenko, Xuan Cao, Runmin Wei, Yuanhao Wu, Xun Zhu, Kuan-Lun Tseng, Zhifeng Gao, Cheng Ju, Xiaohan Yi, Hongdong Zheng, Constantin Kappel, and Emma Lundberg. Analysis of the Human Protein Atlas Image Classification competition. *Nature Methods*, 16(12):1254–1261, 2019.

[27] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[28] Sara Beery, Elijah Cole, and Arvi Gjoka. The iwildcam 2020 competition dataset. In *CVPR Fine-Grained Visual Categorization Workshop (FGVC)*, 2020.

[29] Alistair EW Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 2019.

[30] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[31] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[32] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[33] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi

Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

[34] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[35] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023.

[36] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

[37] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C.H. Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021.

[38] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

[39] Rayan Krishnan, Pranav Rajpurkar, and Eric J. Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, 2022.

[40] Michael Doron, Théo Moutakanni, Zitong S. Chen, Nikita Moshkov, Mathilde Caron, Hugo Touvron, Piotr Bojanowski, Wolfgang M. Pernice, and Juan C. Caicedo. Unbiased single-cell morphology with self-supervised vision transformers. *bioRxiv*, 2023.

[41] Yi Wang, Conrad M Albrecht, Nassim Ait Ali Braham, Lichao Mou, and Xiao Xiang Zhu. Self-supervised learning in remote sensing: A review. *IEEE Geoscience and Remote Sensing Magazine*, 2022.

[42] Caroline Choi, Yoonho Lee, Annie Chen, Allan Zhou, Aditi Raghunathan, and Chelsea Finn. AutoFT: Learning an objective for robust fine-tuning. *arXiv:2401.10220*, 2024.

[43] Helen Qu and Sang Michael Xie. Connect later: Improving fine-tuning for robustness with targeted augmentations. In *ICML*, 2024.

[44] Jeffrey T Leek, Robert B Scharpf, Héctor Corrada Bravo, David Simcha, Ben Langmead, W Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 2010.

[45] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In *Domain adaptation in computer vision applications*, pages 153–171. Springer, 2017.

[46] Aili Shen, Xudong Han, Trevor Cohn, Timothy Baldwin, and Lea Frermann. Contrastive learning for fair representations. *arXiv:2109.10645*, 2021.

[47] Divyat Mahajan, Mohammad Pezeshki, Charles Arnal, Ioannis Mitliagkas, Kartik Ahuja, and Pascal Vincent. Compositional risk minimization. In *ICML*, 2025.

[48] Marin Scalbert, Maria Vakalopoulou, and Florent Couzinié-Devy. Towards domain-invariant self-supervised learning with batch styles standardization. In *ICLR*, 2024.

[49] I. Korsunsky, N. Millard, J. Fan, K. Slowikowski, F. Zhang, K. Wei, Y. Baglaenko, M. Brenner, P. R. Loh, and S. Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature Methods*, 2019.

[50] J. B. Kang, A. Nathan, K. Weinand, F. Zhang, N. Millard, L. Rumker, D. B. Moody, I. Korsunsky, and S. Raychaudhuri. Efficient and precise single-cell reference atlas mapping with symphony. *Nature Communications*, 2021.

[51] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 2018.

[52] Hirofumi Kobayashi, Keith C. Cheveralls, Manuel D. Leonetti, and Loic A. Royer. Self-supervised deep learning encodes high-resolution features of protein subcellular localization. *Nature Methods*, 2022.

[53] Ankit Gupta, Zoe Wefers, Konstantin Kahnert, Jan N. Hansen, Will Leineweber, Anthony Cesnik, Dan Lu, Ulrika Axelsson, Frederic Ballllosera Navarro, Theofanis Karaletsos, and Emma Lundberg. SubCell: Vision foundation models for microscopy capture single-cell biology. *bioRxiv*, 2024.

[54] Yeonkyung Lee, Woojung Han, Youngjun Jun, Hyeonmin Kim, Jungkyung Cho, and Seong Jae Hwang. PRETI: Patient-aware retinal foundation model via metadata-guided representation learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 523–533, 2025.

[55] Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David B. Lobell, and Stefano Ermon. SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery. In *NeurIPS*, 2022.

[56] Colorado J. Reed, Ritwik Gupta, Shufan Li, Sarah Brockman, Christopher Funk, Brian Clipp, Kurt Keutzer, Salvatore Candido, Matt Uyttendaele, and Trevor Darrell. Scale-MAE: A scale-aware masked autoencoder for multiscale geospatial representation learning. In *ICCV*, 2023.

[57] Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, Vivek Natarajan, and Mohammad Norouzi. Big self-supervised models advance medical image classification. In *ICCV*, 2021.

[58] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D. Manning, and Curtis P. Langlotz. Contrastive learning of medical visual representations from paired images and text. In *Machine Learning for Healthcare Conference*, 2022.

[59] Samuel Stevens, Jiaman Wu, Matthew J. Thompson, Jake Campolongo, Chan Hee Song, David Edward Carlyn, Li Dong, Wasila Dahdul, Charles Stewart, Tanya Berger-Wolf, Wei-Lun Chao, and Yu Su. BioCLIP: A vision foundation model for the tree of life. In *CVPR*, 2024.

[60] Vicente Vivanco Cepeda, Gaurav Kumar Nayak, and Mubarak Shah. GeoCLIP: Clip-inspired alignment between locations and images for effective worldwide geo-localization. In *NeurIPS*, 2023.

[61] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One embedding space to bind them all. In *CVPR*, 2023.

[62] Rangel Daroya et al. WildSAT: Learning satellite image representations from wildlife observations. In *ICCV*, 2025.

[63] Rich Caruana. Multitask learning. *Machine Learning*, 1997.

[64] Yujia Bao and Theofanis Karaletsos. Contextual vision transformers for robust representation learning. In *ICML Workshop on Spurious Correlations, Invariance and Stability (SCIS)*, 2023.

[65] Mohamed El Banani, Karan Desai, and Justin Johnson. Learning visual representations via language-guided sampling. In *CVPR*, 2023.

[66] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 2018.

[67] Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *ICLR*, 2021.

[68] Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. In *ICLR*, 2022.

[69] Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv:2511.08544*, 2025.

[70] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv:2508.10104*, 2025.

[71] Peter J Thul, Lovisa Åkesson, Mikaela Wiking, Diana Mahdessian, Aikaterini Geladaki, Hammou Ait Blal, Tove Alm, Anna Asplund, Lars Björk, Lisa M Breckels, et al. A subcellular map of the human proteome. *Science*, 2017.

[72] Peter J Thul and Cecilia Lindskog. The human protein atlas: A spatial map of the human proteome. *Protein Science*, 2018.

[73] Nathan H. Cho, Keith C. Cheveralls, Andreas-David Brunner, Kibeom Kim, André C. Michaelis, Preethi Raghavan, Hirofumi Kobayashi, Laura Savy, Jason Y. Li, Hera Canaj, James Y. S. Kim, Edna M. Stewart, Christian Gnann, Frank McCarthy, Joana P. Cabrera, Rachel M. Brunetti,

Bryant B. Chhun, Greg Dingle, Marco Y. Hein, Bo Huang, Shalin B. Mehta, Jonathan S. Weissman, Rafael Gómez-Sjöberg, Daniel N. Itzhak, Loïc A. Royer, Matthias Mann, and Manuel D. Leonetti. OpenCell: endogenous tagging for the cartography of human cellular organization. *Science*, 375(6585):eabi6983, 2022.

[74] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *AAAI*, 2019.

[75] Anatol Garioud, Sébastien Giordano, Nicolas David, and Nicolas Gonthier. FLAIR-HUB: Large-scale multimodal dataset for land cover and crop mapping. *ISPRS Journal of Photogrammetry and Remote Sensing*, 237:271–300, 2026.

[76] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *IJCV*, 2024.

[77] Théo Moutakanni, Piotr Bojanowski, Guillaume Chassagnon, Céline Hudelot, Armand Joulin, Yann LeCun, Matthew Muckley, Maxime Oquab, Marie-Pierre Revel, and Maria Vakalopoulou. Advancing human-centric ai for robust x-ray analysis through holistic self-supervised learning. *arXiv:2405.01469*, 2024.

[78] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *ICLR*, 2022.

[79] Théo Moutakanni, Camille Couprie, Seungeun Yi, Michael Doron, Zitong S Chen, Nikita Moshkov, Elouan Gardes, Mathilde Caron, Hugo Touvron, Armand Joulin, Piotr Bojanowski, Wolfgang M Pernice, and Juan C Caicedo. Cell-dino: Self-supervised image-based embeddings for cell fluorescent microscopy. *PLOS Computational Biology*, 21(12):e1013828, 2025.

[80] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *NeurIPS*, 2018.

[81] Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. FAMO: Fast adaptive multitask optimization. In *NeurIPS*, 2023.

[82] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, 2020.

[83] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *NeurIPS*, 2021.

[84] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *ICML*, 2022.

[85] Dmitry Senushkin, Nikolay Patakin, Arseny Kuznetsov, and Anton Konushin. Independent component alignment for multi-task learning. In *CVPR*, 2023.

[86] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.

Who Needs Labels? Adapting Vision Foundation Models With the Metadata You Already Have

Supplementary Material

Contents

A.1 Additional Robustness Experiments 16
A.2 Backbone Ablation: SigLIP2 17
A.3 Extended Dataset Overview 17
A.4 Per-Factor Metadata Analysis 19
A.5 Implementation Details 22
A.6 General Experimental Setup 25
A.7 Probing Protocol Details 26
A.8 Baseline Implementation Details 27
A.9 Cross-Dataset Transfer: Protocols and Results 28
A.9.1 HPA → OpenCell 28
A.9.2 FMoW → FLAIR-Hub 29
A.9.3 MIMIC-CXR → CheXpert 30

A.1 Additional Robustness Experiments

Complete-adaptation reproducibility. We repeated the entire adaptation and evaluation pipeline rather than retraining probes around a fixed representation. Table A.1 reports the resulting mean and standard deviation. Table 1 reports the full-high-resolution HPA result; for compute reasons, the 10-run HPA repeat uses the 224^2 setting of Table 6 and obtains 54.7 ± 0.7.

Table A.1: **Reproducibility across complete adaptation runs.** HPA uses the low-resolution setting; its full-high-resolution result is 61.2 in Table 1.

Dataset	Metric	Runs	Mean±std
FMoW	OOD WGA	5	53.3 ± 0.8
MIMIC-CXR	AUROC	5	81.7 ± 0.1
iWildCam	OOD macro-F1	8	52.9 ± 0.8
HPA (224^2)	F1	10	54.7 ± 0.7

Selection ledger. Table 2 in the main text makes the final metadata assignments and selection signals explicit. Candidate signs were first evaluated on held-out validation data; the adaptation checkpoint and probe hyperparameters were likewise chosen by the listed held-out metric. HPA public/private scores are evaluation metrics, not selection metrics.

Task-label-supervised guidance. We compare metadata-only FINO with three supervised adaptation references while retaining the matched DINO+iBOT machinery and frozen-probe evaluation: conventional SupCon; task-label guidance through our prototype objective; and SupCon combined with the selected metadata. For multi-label HPA, SupCon positives share at least one active label, while prototype guidance uses a uniform target over active classes and updates all active prototypes.

FMoW performance ablations. LP-FT [78] mitigates the degradation from direct supervised fine-tuning but remains below frozen probing and FINO. Although LP-FT and FINO both freeze and then unfreeze a backbone, LP-FT warms a task-supervised classifier; FINO warms unlabeled SSL/metadata heads and the patch embedding. Removing FINO's frozen warm-up while keeping the overall adaptation budget fixed reduces WGA to 47.7, confirming the value of the warm-up at a fixed adaptation budget.

Table A.2: **Supervised task-label guidance on FMoW and low-resolution HPA.** Metadata-only FINO is strongest on both OOD metrics.

	FMoW WGA		HPA F1	
Adaptation guidance	IID	OOD	IID	OOD
SupCon (task labels)	72.3	48.3	68.6	54.5
Task-label prototypes	**72.8**	48.5	67.4	53.1
SupCon + selected metadata	72.7	51.0	**69.7**	54.0
FINO (metadata only)	72.1	**53.3**	68.7	**54.7**

Table A.3: **Additional FMoW performance comparisons.** All rows report OOD worst-group accuracy under the matched evaluation protocol.

Method	OOD WGA
Frozen backbone + probe	45.0
Supervised fine-tuning	40.5
LP-FT	44.1
SSL adaptation	48.4
FINO without frozen warm-up	47.7
FINO	**53.3**

Selection and release protocol. All adaptation configurations, checkpoints, metadata assignments, and probe hyperparameters were selected exclusively with held-out validation data. Neither the HPA Kaggle public nor private score informed selection. We release the complete end-to-end code and metadata pipeline, including data acquisition and preprocessing, dataloaders, exact configurations, environments, splits, seeds, probes, metrics, commands, and table scripts. Laboratory policy prohibits redistribution of trained checkpoints.

A.2 Backbone Ablation: SigLIP2

The main paper builds on a DINOv3 [70] initialisation, whose self-distilled features are already strong on the type of dense, fine-grained content emphasised by our benchmarks. A natural concern is therefore whether the gains attributed to FINO actually come from metadata-guided adaptation, or whether they are tied to specifics of the DINOv3 recipe. To probe this, we replicate the entire representation adaptation pipeline on top of a SigLIP2 [3] ViT-L/16 checkpoint, using the publicly available HuggingFace weights[2] and keeping all hyperparameters, heads, and probing protocols identical to the DINOv3 runs (cf. App. A.5 and App. A.7). Tab. 7 in the main text reports each benchmark's primary metric for the off-the-shelf frozen backbone, the $\mathcal{L}_{\text{DINO}}$-only adaptation, and the full FINO objective.

Discussion. On the SigLIP2 backbone, both stages of the adaptation pipeline yield monotonic gains across all four datasets. On iWildCam, self-supervised adaptation alone lifts OOD F1 from 11.3 to 16.9 ($+5.6$), and adding metadata guidance contributes a further $+4.5$, reaching 21.4. The same ordering holds on FMoW worst-group accuracy, where $\mathcal{L}_{\text{DINO}}$ recovers $+5.1$ points over the frozen baseline ($30.0 \rightarrow 35.1$) and FINO adds another $+1.7$ on top ($35.1 \rightarrow 36.8$). The magnitudes of both contributions are comparable to those reported on a DINOv3 initialisation in Tab. 1, despite SigLIP2 being trained with a fundamentally different (image–text contrastive) objective. This supports the interpretation that metadata-guided adaptation acts on top of, rather than in lieu of, a strong visual prior, and that its benefit is largely backbone-agnostic.

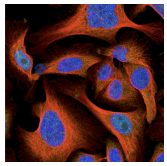
A.3 Extended Dataset Overview

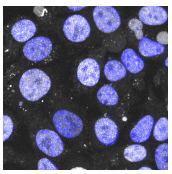
We provide in Fig. A.1 a detailed overview of all datasets used in this paper. We define *micro-domains* as the fine-grained domains within an application domain (e.g., plates in microscopy, countries in Earth observation). In-distribution (ID) and out-of-distribution (OOD) evaluations are defined with respect to these micro-domains, which correspond to the domain shifts typically considered in UDA.

[2] `https://huggingface.co/collections/google/siglip2`

- **Human Protein Atlas (HPA) [71, 72].** A fluorescence microscopy dataset for protein localisation, consisting of four-channel cellular images with strong batch effects and acquisition variability. Micro-domains correspond to *plates*, i.e., groups of images acquired under the same experimental conditions (e.g., staining, illumination, or preparation batch). The source (train) data contains 94,270 images from 1,239 distinct plates. These images correspond to the Kaggle train images and images downloaded from the human protein atlas website as recommended [26]. They match the HPA foV images listed by [79], resized to 768×768 resolution. The target data corresponds to the test set from the Human Protein Atlas Kaggle competition [26] and contains 10,634 images from 218 plates. We report the macro F1-score. The metadata include plate identity, antibody type, and cell line, all discrete.
- **Functional Map of the World (FMoW) [27].** A large-scale Earth observation dataset of satellite images annotated for land use classification. We use the WILDS benchmark split [17], which introduces substantial geographic and temporal distribution shifts. Micro-domains correspond to geographic regions (countries) and acquisition time (year). The source data comprises 76,863 images across 146 countries, while the target data contains 22,108 images spanning 171 countries. We report macro-accuracy and worst-group accuracy, defined as the accuracy on the worst-performing domain. The metadata include both discrete (region, country, year, month) and continuous (hour, coordinates, ground sampling distance, sun elevation, off-nadir angle) factors.
- **iWildCam [28].** A camera trap dataset for wildlife classification, characterised by long-tailed distributions and strong environmental variations across locations. Micro-domains correspond to camera trap locations. The source data contains 129,809 images from 243 locations, while the target data follows the WILDS split with 42,791 images from 48 distinct locations for evaluation. We report the macro F1-score on both the in-distribution (ID) and out-of-distribution (OOD) test sets defined in the WILDS benchmark. The metadata include trap identity (discrete) and timestamp (continuous). We used 336×336 global crops similarly to [42].
- **MIMIC-CXR [29].** A large-scale dataset of de-identified chest radiographs paired with free-text radiology reports, used here for multi-label pathology classification over 14 findings. Unlike the other three primary benchmarks, this dataset has no natural micro-domain partition, so we adopt an in-distribution (IID) protocol with $368{,}960$ training images and $5{,}159$ IID test images. We report the macro AUROC averaged across the 14 pathology labels. The dataset exposes both discrete metadata (radiograph view position, patient sex) and continuous metadata (patient age); FINO relies only on the discrete view position.

[3]The X-ray illustrations are not from MIMIC-CXR nor CheXpert, but a personal Xray from one of the authors.

	HPA	iWildCam	FMoW	MIMIC-CXR [3]
				
Domain	Fluor. microscopy	Wildlife	Earth obs.	Chest X-ray
Micro-domains	plates	traps	countries	— (IID)
Source	94,270 img 1,239 plates	129,809 img 243 traps	76,863 img 146 countries	368,960 img 5 view pos.
Target	23,599 img 218 plates	42,791 img 48 traps	22,108 img 171 countries	5,159 img (IID test)
Metadata used in Tab. 1	••• antibody ($\mathbb{M}_+$)	↔ time of day ($\mathbb{M}_-$)	••• subregion ($\mathbb{M}_+$) ••• year ($\mathbb{M}_-$)	••• view position ($\mathbb{M}_+$)
Other metadata	••• plates ••• cell line	••• trap ••• month	••• region, country, month ↔ hour, lon, lat ↔ sun elev, view angle, res	••• patient sex ↔ patient age

	OpenCell	*FLAIR-Hub*	*CheXpert* [3]
			
Transfer from	HPA	FMoW	MIMIC-CXR
Channels	2: nucleus, protein *(microtubule, ER dropped)*	RGB only *(other bands dropped)*	single channel *(unchanged)*
Resolution	100^2 crops *(vs 768^2 in HPA)*	0.2 m/px aerial *(vs ∼0.3–1.5 m satellite)*	high-res X-ray *(unchanged)*
Labels	subcellular localisation *(clustering protocol)*	15 land-cover classes, segmentation masks *(vs 62 image-level classes in FMoW)*	14 findings, *Stanford vs Beth Israel*

••• discrete ↔ continuous metadata

Figure A.1: **Extended overview of evaluation datasets.** Datasets span Earth observation (FMoW, FLAIR-Hub), wildlife monitoring (iWildCam), fluorescence microscopy (HPA, OpenCell), and medical imaging (MIMIC-CXR, CheXpert). Representative sample images are shown alongside the application domain, micro-domain structure (fine-grained factors of variation such as geographic regions, camera traps, or experimental plates), dataset scale, and available metadata, categorised as discrete (•••) or continuous (↔).

A.4 Per-Factor Metadata Analysis

The compass plot of Fig. 3 sweeps every candidate metadata factor through the two single-branch assignments (adding it in isolation to the encouraged set $\mathbb{M}_+$ or the suppressed set $\mathbb{M}_-$) and records the resulting change Δ in the dataset's primary metric. Each factor's pair of Δs places it in one of four quadrants: *always beneficial* (helps in either branch), *always harmful* (hurts in either), *informative only* (helps in $\mathbb{M}_+$, hurts in $\mathbb{M}_-$), and *spurious only* (the converse). Most factors fall cleanly into *always beneficial*, *informative only*, or *spurious only*. Two HPA factors do not: *plate* surfaces as *informative only*, even though as the canonical carrier of microscopy batch effects it ought to live in *spurious only*; and *cell line* lands in *always harmful* despite carrying obvious task-relevant structure.

Both anomalies trace back to entanglement between candidate factors. App. A.4.1 and App. A.4.2 diagnose each in turn, and App. A.4.3 distills the lesson into a practical guideline.

A.4.1 Plate: "informative only" as a side-effect of two independent leakage pathways

The plate factor lands at $\Delta = +0.31$ when encouraged and $\Delta = -0.51$ when suppressed (Fig. 3), placing it in the *informative only* quadrant. This is surprising: a plate is a physical experimental unit sharing illumination, fixation, and imaging settings, and is the canonical unit of *batch effect* in high-throughput microscopy.

Two independent leakage pathways. The HPA dataset (source plus target) covers $104{,}904$ images, $1{,}457$ plates, $14{,}678$ antibodies, and 35 cell lines. Each plate uses a single cell line (99.25% of plates carry exactly one) and hosts a panel of ~ 25 antibodies (median 25, max 64); each antibody mostly lives on one plate. Tab. A.4 measures these dependencies through the asymmetric uncertainty coefficient $U(Y \mid X) = 1 - H(Y \mid X)/H(Y) \in [0, 1]$, the fraction of Y's entropy removed by knowing X. Two entries dominate: $U(\text{cell line} \mid \text{plate}) = 0.999$ (one cell line per plate) and $U(\text{antibody} \mid \text{plate}) = 0.652$ (each plate selects a ~ 25-antibody panel out of $14{,}678$). The two pathways are independent: controlling for cell line leaves $U(\text{antibody} \mid \text{plate}, \text{cell line}) = 0.652$ unchanged, so plate's predictiveness of antibody is not mediated by cell line. Plate therefore carries two distinct task-relevant axes (the cell substrate and the antibody panel chosen on it) rather than a single nuisance variable.

Table A.4: **Asymmetric dependence between HPA discrete metadata factors.** Each cell reports $U(Y \mid X) = 1 - H(Y \mid X)/H(Y) \in [0, 1]$, the fraction of the entropy of Y removed by knowing X. Cardinalities ($1{,}457$ plates, $14{,}678$ antibodies, 35 cell lines) explain part of the asymmetries: a low-cardinality variable on the column side is easy to "determine" from a high-cardinality one. The two load-bearing entries are $U(\text{cell line} \mid \text{plate}) = 0.999$ (one cell line per plate by construction) and $U(\text{antibody} \mid \text{plate}) = 0.652$ (each plate selects a ~ 25-antibody panel).

$U(Y \mid X)$, $X \downarrow$ $Y \rightarrow$	plate	antibody	cell line
plate	1.000	0.652	0.999
antibody	0.869	1.000	0.567
cell line	0.303	0.129	1.000

Encouragement: plate is a coarsening of both informative axes. Plate guidance can be read in two complementary ways. Through $U(\text{cell line} \mid \text{plate}) = 0.999$, pulling plate-aligned images together pulls same-cell-line images together: the $1{,}457$ plate prototypes split each of the 35 cell-line clusters into ~ 40 sub-clusters. Through $U(\text{plate} \mid \text{antibody}) = 0.869$, the same operation also coarsens antibody guidance, bundling the $14{,}678$ antibodies into groups of ~ 25. Plate's encouragement endpoint sits between those of the two axes it spans: cell line at $\Delta = -1.30$ (the per-cell-line trap of App. A.4.2), antibody at $\Delta = +1.80$, plate at $+0.31$, closer to the antibody side because the finer the partition, the further it pushes the encoder away from collapsing onto cell-line identity. The benefit is real but largely subsumed by antibody when both branches are available.

Suppression: two independent leaks of informative signal. Gradient reversal on plate hits both pathways at once. Through $U(\text{cell line} \mid \text{plate}) = 0.999$, it amounts to gradient reversal on cell line, which erases the cell-morphology substrate against which the four fluorescence channels are read (App. A.4.2). Through the residual $U(\text{antibody} \mid \text{plate}, \text{cell line}) = 0.652$, it also suppresses the antibody-panel structure within each cell line, removing the fine-grained protein-related signal that the antibody branch would otherwise encourage. With no antibody anchor in $\mathbb{M}_+$ to point to, the encoder has no direction left: the desired batch-effect-only signal is the smallest of the three components (cell line, antibody panel, residual batch effect), and the -0.51 endpoint reflects the cost of erasing the other two.

A.4.2 Cell line: a necessary information

Cell line measures $\Delta = -1.30$ when encouraged and $\Delta = -0.52$ when suppressed (Fig. 3), placing it in the *always harmful* quadrant. The label is correct *for the binary* $\mathbb{M}_+/\mathbb{M}_-$ *knob*, but masks two opposing pressures with only a narrow benign band between them:

- *Cell morphology is part of the input substrate.* Protein localisation in HPA is read *against* the cellular context (nucleus, membrane, cytoskeleton); fully unlearning morphology removes the reference frame the four fluorescence channels are interpreted in, so aggressive suppression collapses both the substrate and the main task.
- *Cell morphology correlates with the target.* Different cell lines express different protein repertoires and exhibit systematically different localisation patterns, so encouraging the encoder to organise its features along cell-line identity traps protein representations in a per-cell-line manifold and hurts cross-cell-line generalisation.

The sweet spot retains enough cell morphology to read the image at all, while staying loose enough that protein-localisation features are not pinned to cell type. The binary knob lands on neither side: $\mathbb{M}_+$ overshoots into the per-cell-line trap, $\mathbb{M}_-$ overshoots into substrate erasure, and both endpoints register as harmful.

The UMAPs and the metadata cardinalities corroborate this reading. Fig. 4 shows that cell-line k-NN accuracy is high under self-supervised adaptation but drops to 74% for FINO (with antibody as the only informative guide and $\mathbb{M}_- = \varnothing$), while protein-localisation k-NN climbs from 4% to 83%. Two mechanisms explain why an antibody guide alone lands close to the sweet spot:

- *Cardinality split.* HPA contains $14{,}678$ antibodies for only 35 cell lines, so each cell-line cluster contains on average ~ 420 antibody prototypes. The antibody contrastive loss splits each cell-line cluster into roughly that many sub-clusters and pushes them apart, breaking the coarse cell-line clustering captured by frozen DINOv3. The decoupling is structural: it follows from the partition refinement, not from any specific correlation in the data.
- *Anchor pressure, not collapse.* The antibody guide is the dataset's target-aligned anchor, so it overlaps with the rest of the meaningful structure by design: that is what makes it informative, and what stops it from over-erasing the other axes. The dependence $U(\text{cell line} \,|\, \text{antibody}) = 0.57$ in Tab. A.4 quantifies the effect: pulling same-antibody images together preserves a substantial share of cell-line structure as a by-product, consistent with cell-line k-NN dropping to 74% rather than to chance.

A.4.3 Practical guideline

Both HPA cases above stem from the same issue (entanglement between candidate factors makes the binary $\mathbb{M}_+/\mathbb{M}_-$ assignment misleading), though they manifest differently: plate as a load-bearing aggregate of two informative axes, cell line as a target whose two failure modes the binary knob cannot interpolate between. The diagnostic is itself derived from the metadata table alone, and is computable in ~ 2 minutes from the discrete metadata columns without running a single training job; the compass endpoints are the consistency check on it, not its source. Three takeaways follow:

- *Check the dependence structure first.* Before assigning a candidate factor to $\mathbb{M}_+$ or $\mathbb{M}_-$, compute $U(Y \,|\, X)$ as in Tab. A.4 against the other available factors (or use mutual information / coefficient of determination for symmetric or mixed cases). Conditional variants such as $U(Y \mid X, Z)$ separate direct from mediated dependence, as in the plate-antibody-cell-line analysis above. The full table is a histogram count over the metadata columns; on HPA's $\sim 10^5$-row table it runs in minutes on a CPU, so the test is available *before* any branch assignment is committed to.
- *Prefer positive guidance for entangled factors.* When the dependence test of the previous bullet shows that a candidate factor is a coarsening of a target-aligned axis (here, plate determines cell line at $U = 0.999$ and selects a ~ 25-antibody panel at $U = 0.652$), the binary knob has only one safe setting: assigning it to $\mathbb{M}_+$ recovers most of that signal at no risk of collapse, at the cost of leaving residual batch effects in the representation. This inverts the textbook microscopy reflex of suppressing plate as a pure batch carrier (*e.g.*, Harmony [49], Symphony [50], Scalbert et al. [48]), and the inversion is exactly what the dependence table predicts: the standard reflex is calibrated on the assumption that plate is conditionally independent of the target given the image, which the $0.999/0.652$ entries reject. Plate's $+0.31/-0.51$ compass endpoints are then the empirical confirmation of that prior prediction, not the basis for it.
- *Never suppress an entangled factor in isolation.* Suppression alone is what destabilises the optimisation: the gradient-reversal pressure has no direction to point in. Pairing it with a positive guide on the entangled axis is the natural fix, in the spirit of DANN [18] coupling

a domain-adversarial head to a label-supervised classifier; here the analogue would be to suppress plate jointly with antibody as a $\mathbb{M}_+$ anchor. When the entangled factor is purely nuisance, our recipe already supplies the scaffolding: place the informative variable in $\mathbb{M}_+$ and the nuisance in $\mathbb{M}_-$ in the *same* run, and let the per-branch gradient equalisation of App. A.5 balance the two pressures (as we do on FMoW with sub-region encouraged and year suppressed).

A.5 Implementation Details

Sample minibatch $\{(x_n, \{m_n^{(t)}\}_{t\in\mathcal{T}})\}_{n=1}^{B}$
Compute student $\phi(x_n)$ and teacher $\hat{\phi}(x_n)$ embeddings
Compute $\mathcal{L}_{\text{DINO}}$ and $\mathcal{L}_{\text{iBOT}}$
for each metadata type $t \in \mathcal{T}$ **do**
    if t discrete **then**
        Compute $\mathcal{L}_{\text{meta}}^{(t)}$ via prototypes $\{p_k^t\}$; update prototypes by EMA
    else
        Compute $\mathcal{L}_{\text{meta}}^{(t)}$ via predictor $g^{(t)}$
    end if
    if $t \in \mathbb{M}_-$ **then**
        Apply gradient reversal to encoder
    end if
end for
Equalise per-branch gradient norms by $s_t = \bar{n}/\tilde{n}_t$ (App. A.5)
Apply $\mathcal{L}_{\text{SIGReg}}$ to the pre-normalisation bottleneck
Update student by AdamW and teacher by EMA

Figure A.2: **One training step of FINO.** Informative metadata is encouraged, while spurious metadata is suppressed via gradient reversal.

Input resolution. Wherever a benchmark has an established input resolution from prior work, we adopt it directly: we match Choi et al. [42] on the WILDS benchmarks for both FMoW and iWildCam (336^2 on iWildCam), and use 512^2 on MIMIC-CXR following the standard chest-radiograph protocol. HPA is the one dataset where we deliberately depart from the source resolution: we use 768^2 in the main runs and 576^2 in the SigLIP2 ablation of App. A.2. Both are downscales adopted on *compute* grounds (a single full-resolution HPA run already dominates the per-dataset GPU-hours of Tab. A.6) and not a recipe constraint. The method itself imposes no resolution requirement and operates identically at either size; the two HPA values simply bracket what comfortably fit our compute envelope.

Training schedule. Training proceeds in two phases unless otherwise noted. The first phase is a short frozen-backbone warm-up in which the patch embedding, the DINO and iBOT heads, the metadata guidance modules, and SIGReg on the DINO bottleneck are trained while the rest of the encoder is held fixed; we did not tune its length and the method is robust to that choice. In the second phase the backbone is unfrozen with a linear learning-rate warmup over 1k iterations, after which all parameters are trained jointly. We recommend training for at least 50 epochs and selecting the best-validation checkpoint: this length acts as an upper bound on the search rather than a tuned hyperparameter, since best-validation selection is applied uniformly to every method we compare against and the same envelope is therefore granted to all of them.

iWildCam is the one departure from this two-phase recipe. As a camera-trap dataset of essentially natural images it sits unusually close to the DINOv3 pretraining distribution (closer than any typical adaptation target the method is designed for) and we observed prolonged adaptation to actively erode the pretrained features. We therefore skip the frozen warm-up and cap adaptation at $\sim$12 epochs. This reflects the proximity of iWildCam to DINOv3's pretraining distribution and is not a hyperparameter the method asks the user to tune on more typical adaptation targets.

Heads and remaining hyperparameters. Both DINO and iBOT heads use 65,536 prototypes, a hidden dimension of 2048, a bottleneck dimension of 256, and 3 MLP layers. Discrete metadata prototype branches use a temperature $\tau = 0.023$ and a centroid momentum $\alpha = 0.99$. We use a constant learning-rate schedule after warmup and `float16` mixed precision. The method showed

robustness to τ; the retained value $0.023 = 0.07/3$ is one of three values explored during method development.

ℓ_2 normalisation in the discrete prototype branches. The inner product $\langle \phi(x), p_m^t \rangle$ in Eq. (1) of the main paper is computed as a cosine similarity: both the student embedding $\phi(x)$ and each prototype p_m^t are ℓ_2-normalised onto the unit sphere immediately before the dot product. Consistently, the EMA update of Eq. (2) is applied to the ℓ_2-normalised teacher embedding $\phi_{\text{teacher}}(x)/\|\phi_{\text{teacher}}(x)\|_2$, and the resulting p_m^t is re-projected onto the unit sphere before being used in any subsequent similarity computation.

Tab. A.5 reports the loss weights used to balance the different objectives during training. These values are held fixed across datasets and metadata types: every metadata branch in the main results uses $\lambda_{\text{meta}}^{(t)}$=0.03, regardless of the application domain or whether the branch is discrete or continuous. We adopt λ_{SIGReg}=0.05 as proposed by the original LeJEPA paper [69] and never tune it. The only loss-weight knob we ever sweep is λ_{KoLeo}=0.1, used only by the $\mathcal{L}_{\text{DINO}}$-only baseline runs in place of SIGReg.

Table A.5: **Loss weights used during training.** Each term is scaled by a constant factor in the total objective. The DINO and iBOT loss weights are kept at their defaults (λ_{DINO}=λ_{iBOT}=1) and are not listed. $\mathcal{L}_{\text{KoLeo}}$ is only used for the $\mathcal{L}_{\text{DINO}}$-only baseline runs and is set to 0 whenever $\mathcal{L}_{\text{SIGReg}}$ is active.

Loss	Weight	Value
$\mathcal{L}_{\text{KoLeo}}$	λ_{KoLeo}	0.1
$\mathcal{L}_{\text{SIGReg}}$	λ_{SIGReg}	0.05
$\mathcal{L}_{\text{meta}}^{(t)}$	$\lambda_{\text{meta}}^{(t)}$	0.03

Metadata gradient schedule. The gradient flowing from each metadata branch back into the backbone is modulated by a scalar $\gamma(s) \in [0, 1]$ following the standard DANN ramp [18]:

$$\gamma(s) = \frac{2}{1 + e^{-10\, s/S}} - 1, \tag{5}$$

where s and S are counted from the moment the backbone is unfrozen (i.e. from the Phase 1→Phase 2 transition when a frozen Phase 1 is used). The sign is flipped for spurious factors $t \in \mathbb{M}_-$ (gradient reversal). The schedule applies symmetrically to both metadata branches: continuous heads $g^{(t)}$ keep training with $+1$ on their own parameters, and discrete prototype banks $\{p_k^t\}_{k=1}^K$ keep being updated by EMA from teacher embeddings, so both representations of the metadata structure converge independently of $\gamma(s)$. Only the encoder is shielded: it sees no metadata gradient at the moment it unfreezes and ramps in smoothly, avoiding adversarial pressure against heads or prototypes that have not yet stabilised.

Per-branch gradient equalisation. The fixed weights $\lambda_{\text{meta}}^{(t)}$ of Tab. A.5 encode the relative importance the user assigns to each metadata branch. When two or more branches are trained jointly, that intent is corrupted by the very different raw gradient magnitudes of the underlying losses (prototypical contrastive on discrete factors, ℓ_2 regression on continuous ones), under which a single branch can dominate the encoder's updates. We therefore apply, at every training step, a multiplicative correction s_t that equalises gradient ℓ_2 norms across branches before the fixed $\lambda_{\text{meta}}^{(t)}$ are honoured. With $|\mathcal{T}| = 1$ the target $\bar{n}$ collapses to $\tilde{n}_t$ and $s_t \equiv 1$, so the procedure is a no-op and is only meaningful in the multi-guide regime. It has no learnable parameters and no auxiliary objective.

Per-step update. The full procedure is given in Alg. A.3. At the first step, each $\tilde{n}_t$ is initialised to the raw branch gradient norm rather than to zero or an externally supplied value, so that the EMA starts on the correct scale. Probing at the shared CLS embedding $\phi(x)$ rather than at the encoder parameters is what keeps it cheap: a single `torch.autograd.grad` call returns the branch gradient norm without an extra full backward pass through the FSDP-sharded backbone. This relaxation, of using feature-level rather than parameter-level gradients as a surrogate for the shared encoder, was previously used in IMTL-G [67] and RotoGrad [68] for the same reason.

Behaviour in practice. The scales s_t are recomputed every step from the live EMAs and drift slowly toward unity as the EMAs stabilise. On FMoW, for instance, s_t for the `sub_region` and `year`

Require: metadata branches $\mathcal{T}$; EMA decay $\mu = 0.99$; fixed weights $\{\lambda^{(t)}_{\text{meta}}\}_{t\in\mathcal{T}}$; running norms $\{\tilde{n}_t\}_{t\in\mathcal{T}}$

for each branch $t \in \mathcal{T}$ **do**
  $\tilde{n}_t \leftarrow \mu\,\tilde{n}_t + (1-\mu)\left\|\nabla_{\phi(x)}\,\mathcal{L}^{(t)}_{\text{meta}}\right\|_2$ ▷ EMA of gradient norm at the CLS embedding
end for
$\bar{n} \leftarrow \left(\prod_{t\in\mathcal{T}} \tilde{n}_t\right)^{1/|\mathcal{T}|}$ ▷ geometric-mean target across branches
for each $t \in \mathcal{T}$ **do**
  $s_t \leftarrow \texttt{detach}\big(\bar{n}/\tilde{n}_t\big)$ ▷ closed-form scale; no second-order gradient
end for
return $\sum_{t\in\mathcal{T}} s_t\,\lambda^{(t)}_{\text{meta}}\,\mathcal{L}^{(t)}_{\text{meta}}$ ▷ balanced metadata loss for this step

Figure A.3: **Per-step gradient equalisation across metadata branches.** For each active branch t, an EMA $\tilde{n}_t$ of the gradient ℓ_2 norm is maintained at the CLS embedding $\phi(x)$. The branch loss is rescaled by $s_t = \bar{n}/\tilde{n}_t$, with $\bar{n}$ the geometric mean of the smoothed norms; the fixed weights $\lambda^{(t)}_{\text{meta}}$ of Tab. A.5 are applied on top, separating user-specified relative importance (λ) from runtime scale calibration (s). The detach prevents second-order gradient flow through the scale.

branches start out about 12% apart and converge to ≈ 1 within a few hundred steps, evidence that the correction is genuinely tracking online imbalances rather than acting as a precomputed constant.

Relation to multi-task gradient balancing. The metadata branches play, with respect to the shared encoder, a role analogous to tasks in multi-task learning, and the design space of per-task gradient balancers has been extensively studied in that setting. Methods broadly fall into two families: (i) *magnitude balancing*, which rescales each task loss so that gradient magnitudes are comparable on the shared backbone (GradNorm [66], IMTL-G [67], RotoGrad's magnitude block [68], MGDA [80], FAMO [81]); and (ii) *direction balancing*, which modifies update directions to reduce inter-task conflict (PCGrad [82], CAGrad [83], NashMTL [84], Aligned-MTL [85], RotoGrad's rotation block [68]). Our procedure belongs to the first family: it neither projects gradients onto cones nor solves an inner optimisation over directions, and is therefore conceptually orthogonal to the projection-based methods (PCGrad, CAGrad, NashMTL, Aligned-MTL). Within the magnitude-balancing family, three closed-form methods are directly comparable; we contrast them below.

- *Vs. GradNorm [66].* GradNorm treats the loss weights w_i as *learnable* parameters and updates them by SGD on a separate gradient loss $\mathcal{L}_{\text{grad}} = \sum_i \left|G^{(i)}_W - \bar{G}\cdot r_i^{\alpha}\right|_1$, with $\bar{G}$ the average gradient norm, r_i a relative inverse training rate, and α a tunable asymmetry hyperparameter. The learned w_i are renormalised to $\sum_i w_i = T_{\text{tasks}}$ at each step. Our $\lambda^{(t)}_{\text{meta}}$ are fixed; nothing is learned, α is absent, and there is no sum constraint. We apply the closed-form ratio $s_t = \bar{n}/\tilde{n}_t$ as a multiplicative correction *on top of* the fixed user weight, so that "relative importance" (the table value λ) and "scale calibration" (the live s) are separated rather than absorbed into a single learned coefficient.
- *Vs. IMTL-G [67].* Like ours, IMTL-G is closed-form, hyperparameter-free, and computed at the shared feature. The two methods differ in target: IMTL-G solves for the unique $\{\alpha_t\}$ such that the aggregated gradient has *equal projections* onto each unit task gradient (the angle bisector of $\{g_t\}$), whereas we equalise the smoothed *magnitudes* of $\{g_t\}$ to their geometric mean. IMTL-G uses instantaneous per-batch gradients with no temporal smoothing; we maintain an EMA $\tilde{n}_t$, which absorbs the high variance of branch gradient norms in early training without biasing the long-run scale. IMTL-G also reparameterises the loss weights ($\sum_t \alpha_t = 1$, with α_1 derived from $\alpha_{2:T}$), whereas we leave the user weights $\lambda^{(t)}_{\text{meta}}$ untouched and apply s_t on top.
- *Vs. RotoGrad's magnitude block [68].* RotoGrad equalises per-task gradient norms to a common target $C = \sum_k \alpha_k \|G_k\|$ with $\alpha_k \propto \|G_k\|/\|G^0_k\|$, i.e. a convergence-weighted *arithmetic* mean against the *initial* gradient norms $\|G^0_k\|$. We instead use the *geometric* mean of an *EMA* of current norms: this avoids both the dependence on a single early-training reference (which can be unrepresentative under our two-phase schedule with a frozen backbone) and the need to rank tasks by relative convergence. RotoGrad additionally introduces learnable per-task rotation matrices R_k to align directions; our procedure has no such learnable component and addresses magnitude only.

We do not claim a categorical superiority over these methods (we have not run head-to-head experiments swapping them in for our scheme), but rather a different, lightweight point in the design space,

motivated by the specifics of metadata-guided pretraining: a small number of branches ($|\mathcal{T}| \leq 3$ in our experiments), strong heterogeneity between contrastive (discrete) and regression (continuous) losses, and a desire to keep the user-facing loss weights $\lambda_{\text{meta}}^{(t)}$ as a single, interpretable knob for relative importance.

Expanded SIGReg We apply SIGReg [69] to the pre-normalisation bottleneck features z_n produced by the DINO head MLP. The loss matches the empirical characteristic function of projected features to that of an isotropic Gaussian:

$$\mathcal{L}_{\text{SIGReg}} = \frac{1}{|A|} \sum_{a \in A} \int \left| \frac{1}{N} \sum_{n=1}^{N} e^{i\omega\, a^\top z_n} - e^{-\omega^2/2} \right|^2 e^{-\omega^2/2}\, d\omega \,, \tag{6}$$

where A is a set of random unit directions, N is the batch size, and $e^{-\omega^2/2}$ is the characteristic function of $\mathcal{N}(0,1)$. The original SIGReg formulation uses a dedicated projection head; we instead reuse the DINO head bottleneck and apply the loss before ℓ_2 normalisation and prototype projection. All other hyperparameters (number of slicing directions, frequency sampling, and integration scheme), as well as the loss weight λ_{SIGReg}=0.05, are taken directly from [69] and never tuned per dataset.

A.6 General Experimental Setup

Backbone initialisation. All experiments use a ViT-L/16 backbone (306M parameters) initialised from the open-source DINOv3 [70] checkpoint released by the authors,[4] obtained by knowledge distillation from a ViT-7B teacher trained on the LVD-1689M dataset, providing a strong general-purpose visual initialisation for all downstream adaptation and fine-tuning. This same checkpoint is used in all reported settings, including supervised fine-tuning, domain adaptation, representation adaptation, and frozen probing (e.g., the *frozen model* and *pretrained* rows of Tab. 1); the only differences across rows lie in the training objective and which parameters are updated.

Four-channel adaptation for fluorescence microscopy. The HPA dataset consists of four-channel fluorescence microscopy images, while the pretrained DINOv3 checkpoint operates on three-channel RGB inputs. To bridge this gap, we adapt the patch embedding layer. Let $\boldsymbol{W} \in \mathbb{R}^{d \times 3 \times p \times p}$ denote the pretrained weights (d=1024, p=16). We compute the channel-wise mean $\boldsymbol{m} = \frac{1}{3} \sum_{c=1}^{3} \boldsymbol{W}_{:,c,:,:} \in \mathbb{R}^{d \times 1 \times p \times p}$ and form the four-channel embedding:

$$\boldsymbol{W}' = \tfrac{3}{4} \left[\boldsymbol{W} \parallel \boldsymbol{m} \right] \in \mathbb{R}^{d \times 4 \times p \times p}, \tag{7}$$

where $\parallel$ denotes channel-wise concatenation. The factor $3/4$ preserves the expected activation magnitude when summing over four input channels instead of three. All remaining backbone weights are kept unchanged.

Architecture of predictor $g^{(t)}$. For each continuous metadata type t, $g^{(t)}$ is a 3-hidden-layer MLP applied to the backbone embedding $\phi(x) \in \mathbb{R}^d$ (d=1024). Each hidden block is $\text{Linear} \to \text{GELU} \to \text{Dropout}(0.5)$, with widths $[512, 512, 256]$, followed by a linear projection $\mathbb{R}^{256} \to \mathbb{R}^{n_t}$ and an output activation $\sigma_t \in \{\text{sigmoid}, \tanh, \text{id}\}$ matched to the target range. Cyclic factors (hour) are encoded as $(\sin, \cos)$ pairs (n_t=2), geographic coordinates as (lat, lon) (n_t=2), and scalar bounded factors (cloud cover, sun elevation, off-nadir angle, ground sampling distance) as n_t=1. Targets are linearly mapped per-dimension via $(m_{\min}^{(t)}, m_{\max}^{(t)})$ into the codomain of σ_t ($[0,1]$ for sigmoid, $[-1,1]$ for tanh), and the regression loss is computed in this normalised space.

Stability of gradient reversal on continuous targets. With prediction and target both confined to the codomain of $\sigma_t \in \{\text{sigmoid}, \tanh\}$, the squared error is bounded above and the reversed gradient saturates as $g^{(t)}$ approaches the worst-case constant predictor on that interval. SIGReg on the DINO bottleneck (App. A.5) bounds the projected feature norm, and the DANN ramp $\gamma(s)$ of App. A.5 keeps the reversed gradient at zero during the frozen-backbone warm-up. Identity activation is reserved for factors not under reversal and never paired with $t \in \mathbb{M}_-$.

[4] `https://huggingface.co/facebook/dinov3-vitl16-pretrain-lvd1689m`

Compute resources. All experiments were conducted on H100 or H200 GPUs in a dedicated cluster. Tab. A.6 reports the cumulative GPU-hours spent on each dataset, broken down between backbone adaptation (*pretraining*) and downstream probing (*eval*). These numbers reflect the *total* compute consumed by the project, including ablations, preliminary experiments, and method development, rather than only the runs reported in the paper. For comparison, Tab. A.7 isolates the cost of the single final FINO run used to produce the Tab. 1 numbers on each dataset, which together account for less than 15% of the project total. Both tables are derived under the conservative assumption of constant 100% GPU utilisation for the full duration of every job, which substantially overestimates actual energy use. Transfer datasets (OpenCell, CheXpert, FLAIR-Hub) are not listed as they involve only a handful of probing runs and contribute negligibly compared to the figures reported here.

Table A.6: **Cumulative GPU-hours per dataset and stage.** Totals aggregate all jobs run during method development, including ablations and preliminary experiments, and assume constant 100% GPU utilisation throughout each job.

Stage	HPA	FMoW	iWildCam	MIMIC-CXR	Total
Pretraining	15,360	2,560	160	11,520	29,600
Eval	97	256	4,000	720	5,073
Total	15,457	2,816	4,160	12,240	34,673

Efficiency. Self-supervised and metadata-guided training introduce a modest overhead relative to standard supervised training due to the use of a teacher model. In practice, this overhead remains small: supervised training runs at 0.68 s/iter, pure self-supervision at 0.74 s/iter ($\sim$9% slower), and FINO at 0.76 s/iter ($\sim$12% slower than supervised training).

Table A.7: **GPU-hours of the final FINO runs reported in Tab. 1.**

Stage	HPA	FMoW	iWildCam	MIMIC-CXR	Total
Pretraining	3,840	160	8.0	896	4,904
Eval	15.5	0.8	2.7	24	43.0
Total	3,855.5	160.8	10.7	920	4,947

A.7 Probing Protocol Details

To evaluate the quality of learned representations, we add lightweight classification heads on top of the backbone. For representation adaptation models, we train only the head while keeping the pre-trained backbone frozen. For a given dataset, all baselines use the same head architecture and hyperparameter grid.

Linear head. The CLS tokens from the last four transformer blocks are concatenated with the average-pooled patch tokens from the final block. A single linear layer then maps this vector to class logits. The classifier is trained with AdamW and a cosine annealing schedule; we perform a grid search over learning rate and weight decay. For iWildCam, we report the mean across two runs to account for training variability.

Attentive head. Patch tokens from the last four transformer blocks are concatenated along the feature dimension, and the resulting sequence of dimension $4d{=}4096$ is projected to a lower-dimensional embedding space ($d_{\text{emb}}{=}512$) to reduce the number of trainable parameters. A single cross-attention layer with 8 heads and one learnable query token produces a 512-dimensional vector, which is passed through a linear head. The classifier is trained with AdamW and cosine annealing. For HPA, we use binary cross-entropy loss for multi-label classification. We perform a grid search over learning rate, weight decay, attention dropout, batch size, and the use of curriculum learning, where during the first epoch only single-labelled images are shown to the model. In the case of HPA where there is no official validation set, we use 5-fold cross-validation with splits defined by cell line for robust hyperparameter selection.

A.8 Baseline Implementation Details

All supervised and domain adaptation baselines share the same backbone (ViT-L/16) and classification head, differing only in the adaptation loss and its hyperparameters.

Classification head. All baselines use the same attentive pooling head as FINO (cf. App. A.7), ensuring that performance differences reflect the adaptation strategy rather than the probe. Patch tokens from the last four transformer blocks are concatenated along the feature dimension ($4d$=4096), projected to d_{emb}=512, and pooled by a single learnable query through cross-attention (8 heads), followed by a linear map to the target label space. The backbone CLS token is not used by the head; pooling operates purely over spatial patch features.

Augmentation. For FMoW and iWildCam, we use DINO-style augmentation: random resized crop, horizontal flip, color jitter, random grayscale, and Gaussian blur, with label smoothing of 0.1. For HPA, a domain-specific pipeline [79] self-normalises each fluorescence channel independently, using intensity statistics computed from that channel of the same image.

Optimiser and schedule. All baselines use AdamW (weight decay 0.04, gradient clipping 3.0) with a cosine learning rate schedule. When fine-tuning from the pretrained checkpoint, the peak learning rate is 5×10^{-5} with a 5-epoch warmup and the backbone frozen for the first 5 epochs. All runs train for up to 300 epochs with evaluation every 10 epochs and early stopping (patience 3). Model selection uses the best validation OOD metric: worst-group accuracy for FMoW, macro F1 for iWildCam and HPA.

Domain adaptation methods. We evaluate three adaptation strategies, each governed by a single scalar hyperparameter controlling adaptation strength:

- *CORAL* aligns source and target feature covariance matrices, with loss weight λ held constant during training. The covariance discrepancy is normalised by $1/4d^2$.
- *DANN* trains a 2-layer MLP domain discriminator (hidden dimension 1024) via gradient reversal, with adversarial weight λ following a sigmoid warm-up schedule.
- *Group DRO* reweights per-group training losses via exponentiated gradient ascent with step size η. Groups are defined by metadata: year (16 groups) on FMoW, camera location (243) on iWildCam, cell line (35) on HPA, and view position (5 groups) on MIMIC-CXR. The MIMIC-CXR grouping matches the metadata factor used by the best-performing FINO run on this dataset.

CORAL and DANN additionally leverage unlabelled OOD samples as target data; Group DRO operates on source data only.

Loss weight sensitivity. Tab. A.8 reports performance across the full loss weight sweep for each method. CORAL is robust to λ on FMoW and HPA but degrades sharply on iWildCam at larger values (λ=1.0: 24.6 OOD F1 vs. 38.5 at λ=0.01). DANN shows moderate sensitivity, with the optimal weight varying across datasets. Group DRO exhibits the most dataset-dependent behaviour, and no single step size η dominates across benchmarks: on FMoW, performance is non-monotonic, with WGA peaking at 46.4 for η=0.1 and falling off to 44.2 and 43.7 at η=0.01 and η=1.0 respectively; on HPA with cell line groups, performance is essentially flat across the sweep ($52.6 \rightarrow 52.1 \rightarrow 52.5$ Private F1); on MIMIC-CXR with view-position groups, AUROC drifts down moderately from 80.0 at η=0.01 to 77.7 at η=1.0; and on iWildCam, large steps trigger near-collapse, with OOD F1 falling from 26.1 at η=0.01 to 9.3 at η=1.0. The optimal η therefore depends on both the granularity and the semantics of the chosen grouping: coarse, balanced partitions tolerate or even benefit from aggressive reweighting, whereas fine-grained, long-tailed partitions such as iWildCam camera locations (many with very few examples) destabilise the exponentiated-gradient updates and require small step sizes to avoid collapse. This per-dataset tuning requirement contrasts with FINO, which uses a single set of loss weights across all benchmarks (Tab. A.5).

Alternative DRO grouping strategies. On FMoW, using geographic region (6 groups) instead of year yields lower WGA, with the best region run reaching 44.6% at η=1.0 versus 46.4% at η=0.1 with year groups. On HPA, grouping by plate identity (1,457 groups) underperforms cell line (35 groups) on the Kaggle private F1: the best plate run reaches 50.8% at η=0.1 versus 52.6% for cell

Table A.8: **Loss weight sensitivity for domain adaptation baselines.** All methods use a DINOv3-pretrained ViT-L/16 backbone. We sweep the loss weight λ (CORAL, DANN) or step size η (Group DRO) and report performance: worst-group accuracy (WGA) on the FMoW test set, macro F1 on the iWildCam OOD test set, macro F1 on the HPA Kaggle private test set, and AUROC on MIMIC-CXR. Group DRO uses year groups on FMoW (16 groups), camera locations on iWildCam (243), cell line on HPA (35), and view position on MIMIC-CXR (5); see text for alternative groupings. CORAL and DANN require unlabelled target data, which is unavailable in our IID MIMIC-CXR setting (denoted by —). Italicised column headers denote each benchmark's primary ranking metric.

		FMoW	iWildCam	HPA	MIMIC-CXR
Method	Weight	*WGA*	*OOD F1*	*Private F1*	*AUROC*
CORAL	$\lambda = 0.01$	43.8	38.5	40.5	—
	$\lambda = 0.1$	45.9	33.0	41.1	—
	$\lambda = 1.0$	42.2	24.6	37.8	—
DANN	$\lambda = 0.1$	45.2	38.5	50.2	—
	$\lambda = 0.5$	46.2	33.1	30.5	—
	$\lambda = 1.0$	45.8	36.3	39.2	—
Group DRO	$\eta = 0.01$	44.2	26.1	52.6	80.0
	$\eta = 0.1$	46.4	22.4	52.1	78.7
	$\eta = 1.0$	43.7	9.3	52.5	77.7

line at η=0.01, and η=1.0 further degrades plate to 42.3% while cell line remains stable at 52.5%. This confirms that DRO performance depends on both group granularity and semantic relevance; overly fine-grained partitions fragment the optimisation and prevent effective robust learning.

iWildCam convergence sanity-check. The OOD numbers reported on iWildCam by the from-scratch supervised baseline (test OOD F1 around 10%) are noticeably lower than on FMoW or HPA, so it is worth checking that this baseline is not simply under-trained. Fig. A.4 indicates that this is unlikely: over 300 epochs the training loss falls steadily from 5.2 to 1.0 and training accuracy reaches 93%, yet held-out F1 saturates much earlier: OOD splits flatten around 10% by roughly 100 epochs, while ID splits keep improving only marginally, reaching 29% on test and 26% on validation by the final epoch. This pattern is consistent with the difficulty of the iWildCam distribution shift, as previously documented in the WILDS benchmark [17], rather than with an optimisation issue on our side.

A.9 Cross-Dataset Transfer: Protocols and Results

A.9.1 HPA → OpenCell

Evaluation protocol. OpenCell [73] associates each image with a target protein. Following Kobayashi et al. [52], representations are first averaged at the protein level, then clustered against ground-truth subcellular localisation labels. We report two unsupervised clustering metrics, the Adjusted Rand Index (ARI) and the V-Measure, both bounded in $[0, 1]$ with higher values indicating better agreement with the ground-truth partition. To complement these clustering metrics, which are the only ones reported in the prior work, we additionally train an attentive probe on protein-level embeddings and report its macro F1 score on a held-out test split. Means and standard deviations are computed over 5 clustering seeds; CYTOSELF and SUBCELL numbers are taken from Kobayashi et al. [52] and Gupta et al. [53] respectively.

Discussion. A first observation is that the off-the-shelf DINOv3 backbone, with no adaptation at all, already clears both prior published methods on the metrics they report: Frozen DINOv3 reaches ARI 0.466 vs. 0.339 for SUBCELL and 0.262 for CYTOSELF, and V-Measure 0.652 vs. 0.570 and 0.500 respectively. The genuine FINO contribution on top of that strong starting point is $+0.122$ ARI, $+0.088$ V-Measure and $+12.6$ F1 over Frozen DINOv3.

The ranking among the three adaptation regimes is not consistent across metrics: on the clustering metrics (ARI, V-Measure) supervised adaptation comes second behind FINO, with self-supervised adaptation third; on the attentive-probe F1, the order of the two intermediate baselines flips, with self-

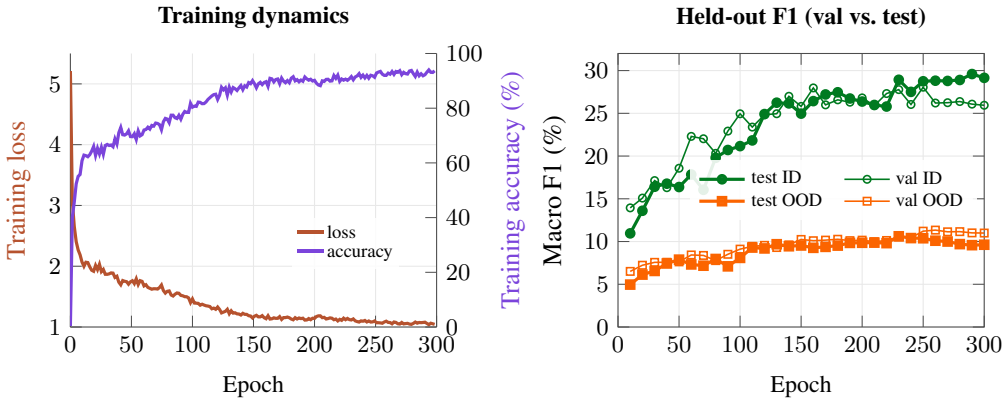


Figure A.4: **iWildCam from-scratch supervised baseline: training behaviour vs. held-out F1.** *Left:* training loss (brown, left axis) and training accuracy (purple, right axis) over 300 epochs of from-scratch supervised training of a ViT-L/16 backbone; loss decreases from ∼5.2 to ∼1.04 and accuracy rises to ∼93%, indicating steady fit to the source distribution. *Right:* macro F1 on the held-out validation and test splits, broken down by in-distribution (green, ID) and out-of-distribution (orange, OOD) micro-domains; thicker lines with filled markers denote the test split and thinner lines with hollow markers the validation split. OOD F1 saturates near ∼10% after roughly 100 epochs, whereas ID F1 keeps improving only marginally, reaching ∼29% (test) and ∼26% (validation) at epoch 300. The low OOD numbers thus reflect the difficulty of the distribution shift rather than under-training. Training is reported in *accuracy*, while held-out evaluation uses the official WILDS macro *F1* metric.

Table A.9: **Full transfer results on HPA → OpenCell.** ARI and V-Measure are computed on protein-level embeddings against ground-truth subcellular localisation labels, following the CytoSelf protocol; F1 is the macro F1-score of an attentive probe trained on protein-level embeddings. The off-the-shelf DINOv3 backbone alone already exceeds both prior published methods on the metrics they report (+0.204 ARI over CytoSelf, +0.127 over SubCell); FINO adds a further +0.122 ARI, +0.088 V-Measure and +12.6 F1 on top of that strong starting point, despite never being trained on OpenCell data. Dashes indicate metrics not reported by the corresponding work.

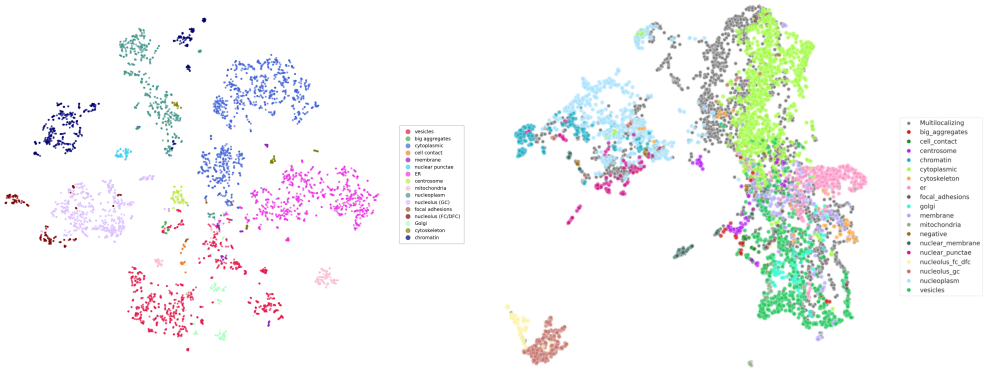
Model	ARI ↑	V-Measure ↑	F1 ↑
CytoSelf [52]	0.262	0.500	—
SubCell [53]	0.339	0.570	—
Frozen DINOv3	$0.466_{\pm 0.019}$	$0.652_{\pm 0.006}$	65.7
Self-supervised adaptation	$0.486_{\pm 0.022}$	$0.668_{\pm 0.006}$	78.0
Supervised adaptation	$0.530_{\pm 0.010}$	$0.710_{\pm 0.006}$	69.2
FINO	$\mathbf{0.588}_{\pm 0.024}$	$\mathbf{0.740}_{\pm 0.010}$	**78.3**

supervised adaptation second and supervised adaptation last (and even below the frozen backbone). The gap between clustering and probe performance for the supervised baseline (high ARI but lower F1) suggests that supervised adaptation tightens clusters around the HPA label set but transfers less well to a probe trained on OpenCell labels. Self-supervised adaptation, conversely, yields a more diffuse but better-transferring representation. FINO is the only regime that wins on all three metrics simultaneously, combining the most structured clusters with the most probe-friendly features. Fig. A.5 visualises this gap qualitatively against the UMAP published by Gupta et al. [53].

A.9.2 FMoW → FLAIR-Hub

We probe the FMoW-adapted backbone on FLAIR-Hub [75], a large-scale aerial imagery benchmark for land-cover and crop mapping at 0.2 m/px (vs. ∼0.3–1.5 m for FMoW satellite imagery). Only the RGB channels are kept to match the FMoW input space; the additional spectral bands are discarded.

DPT head. We use the DPT architecture from Ranftl et al. [86]. The last four transformer layers are extracted, and `CLS` token information is folded into the spatial features via a learned projection



(a) FINO (ours)

(b) SubCell [53], reproduced from original paper

Figure A.5: **UMAP of OpenCell protein-level embeddings, coloured by subcellular localisation.** (a) Our FINO representation. (b) The corresponding visualisation from Gupta et al. [53], shown with their own colour scheme; colour assignments differ between the two panels and are therefore not directly comparable label-by-label. The point of comparison is cluster geometry: FINO produces visibly tighter, better-separated localisation clusters, consistent with the gap in clustering metrics reported in Tab. A.9.

(concatenation followed by a linear layer). Each layer's features are projected to intermediate channel dimensions $[128, 256, 512, 1024]$ with 1×1 convolutions, then spatially aligned to four different scales using transposed convolutions ($4\times$ and $2\times$ upsampling), identity, and a strided 3×3 convolution ($2\times$ downsampling). All four feature maps are then projected to 256 channels with 3×3 convolutions, and progressively fused from coarse to fine through four fusion blocks. Each fusion block consists of two pre-activation residual units followed by $2\times$ bilinear upsampling and a 1×1 projection. A final 3×3 convolution produces the fused representation, which is mapped to class logits by a 1×1 convolution.

A.9.3 MIMIC-CXR $\rightarrow$ CheXpert

We probe the MIMIC-CXR-adapted backbone on CheXpert [74], a chest radiograph dataset acquired at a different institution (Stanford Hospital) than MIMIC-CXR (Beth Israel Deaconess Medical Center), keeping the same multi-label pathology classification task.

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: We make two central claims. First, that metadata can be used to adapt a generic foundation model to a new application domain. Second, that this form of label-free adaptation yields representations that are both more robust and more general than standard label-driven adaptation. The first claim is supported by the overall performance gains over standard adaptation approaches, and further analyzed in Fig. 3, which measures the impact of individual metadata factors. The second claim is supported by the results in Tab. 1, where our method outperforms matched adaptation baselines and obtains higher reported scores than selected specialized systems on three domains while matching the fourth, under differing cross-paper protocols, using only a lightweight probe. We further demonstrate improved generality in Tab. 5, where representations adapted without labels transfer across datasets within the same domain better than supervised adaptation. Finally, Fig. 5 shows that our representations remain expressive across both low- and high-supervision regimes.

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed in the Conclusion and its Broader impact paragraph, including the need to distinguish informative, spurious, redundant, and entangled metadata factors, and potential bias from sensitive or imbalanced metadata.

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 does not include theoretical results or formal proofs; all contributions are empirical.

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: We provide detailed training procedures, hyperparameters, and a step-by-step algorithmic description in the supplementary material. In addition, we release the complete end-to-end code (data and metadata acquisition, preprocessing, training, probing, and evaluation); laboratory policy prevents redistribution of trained checkpoints.

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: All datasets are existing benchmarks. Our public repository provides the end-to-end code for data and metadata download, preprocessing, training, probing, and evaluation, with configurations and commands; the supplementary material documents the protocols. Laboratory policy prevents redistribution of trained checkpoints.

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: The paper specifies all key experimental details, including datasets, evaluation protocols, and model components in the main text. Full training details, hyperparameters, and implementation specifics are provided in the supplementary 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: For experiments involving stochastic label subsampling (low-supervision regime), we report standard deviations across different label samplings (Fig. 5). For the main results, FINO reports mean±standard deviation over complete adaptation repeats (8, 5, and 5 runs on iWildCam, FMoW, and MIMIC-CXR; 10 low-resolution runs on HPA), see Tab. 1, Tab. 6, and the supplementary material.

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: We provide a detailed breakdown of compute resources in the supplementary material, including development and final trainings.

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 complies with the NeurIPS Code of Ethics. All datasets used are publicly available and commonly used in the literature, and no sensitive or personally identifiable information is processed. We also discuss potential biases and ethical considerations related to metadata usage in the paper.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The Broader impact paragraph discusses the main risk, namely that metadata guidance may reproduce or amplify biases encoded by sensitive or imbalanced attributes, and recommends careful metadata selection and auditing beyond aggregate performance. The positive impact, improved label-free adaptation of foundation models to scientific domains, is the subject of the paper.

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 paper does not introduce or release models or datasets that pose a high risk of misuse. It relies on standard vision models and publicly available datasets, and therefore does not require specific safeguards for controlled release.

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 models used in this work are publicly available and properly cited. We follow their respective terms of use and licenses, as described in the original publications.

13. **New assets**

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

    Answer: [Yes]

    Justification: The released code is documented in the public repository, with instructions, configurations, and commands for every stage; trained checkpoints are not redistributed.

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 crowdsourcing or experiments with human subjects. All datasets used are publicly available and were collected by their original authors.

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: The paper does not involve crowdsourcing or research with human subjects. All datasets used are publicly available and no new data collection was performed.

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 were used only for editing purposes and do not contribute to the core methodology or results of the paper.