Does Cross-Panel Pretraining Transfer Under Marker Heterogeneity? A Large-Scale Empirical Study in Clinical Flow Cytometry

Zixin Zhuang[1] **Benjamin S. Mashford**[1,2] **T. Daniel Andrews**[1,2]

[1]School of Computing [2]John Curtin School of Medical Research
Australian National University
{Zora.Zhuang, Benjamin.Mashford, Dan.Andrews}@anu.edu.au

Abstract

Flow cytometry is the primary assay for single-cell immune profiling, yet cross-study modelling remains under-explored. The challenge is two-fold: Studies measure different marker panels, inducing feature-space inconsistency, while batch effects shift signals even on shared markers, making transfer harder than in vision or language where input spaces are stable. We conduct a large-scale empirical study, pretraining an encoder on 89,324 samples from 112 public studies spanning 525 distinct marker panels. We evaluate it frozen — isolating representation quality from task-specific adaptation, on five clinical tasks under a taxonomy of panel-overlap conditions, including three panel-zero-shot tasks whose marker panels were unseen during pretraining, and one negative control with no expected discriminative signal. We further characterise transfer across scaling axes and probe robustness to marker removal at test time. Despite heterogeneity that disrupts direct transfer, the frozen encoder matches or exceeds in-distribution specialist encoders on 4 of 5 tasks, including all three novel-panel tasks, and behaves as expected on the negative control. Scaling analyses reveal saturation in both encoder capacity and random corpus subsampling, and representations degrade gracefully under marker reduction. These results suggest a pretrained encoder can serve as a general-purpose feature extractor for clinical cytometry data, without task-specific retraining. Code, corpus information and model weights are available at `https://doi.org/10.5281/zenodo.23787478`.

1 Introduction

Flow cytometry is a high-throughput single-cell assay for characterising immune cell populations, used routinely in clinical diagnostics and immunological research. For each cell, it records the expression of a panel of cellular markers (surface proteins and/or intracellular signalling molecules) using fluorophore-conjugated antibodies, yielding a marker expression vector per cell with dimensionality typically ranging from 4 to 50, the choice of markers varying across studies.

Public repositories including ImmPort [1] and FlowRepository [2] now contain tens of thousands of samples from hundreds of studies spanning diverse diseases and immune phenotyping contexts. However, this aggregated resource is largely unexploited, even though individual clinical datasets are typically small (tens to low hundreds of samples) yet carry complex, noisy phenotypic signals, making cross-study transfer an attractive modelling paradigm — one that has proven highly effective in language and vision [3, 4, 5]. The barrier is representational. Cross-study cytometry data exhibits

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

two compounding forms of heterogeneity: inconsistent feature spaces from diverse marker panel compositions, and non-biological signal shifts across even shared markers due to batch effects that stem from variations in reagents, instruments, and lab protocols [6]. Flow cytometry data is therefore both noisy and fundamentally partially observed — distinct from other single-cell modalities such as scRNA-seq, where the assay is designed for whole-transcriptome capture, providing a consistent, genome-wide feature space that is well-suited for leveraging large-scale pretrained models [7, 8, 9].

With cross-panel capable architectures and self-supervised pretraining methods recently proposed [10, 11, 12] for learning robust cytometry representations, it is now feasible to pretrain on panel-heterogeneous data at scale. Early results are encouraging, but whether pretrained representations transfer to unseen studies and unseen panels under inherent data heterogeneity, rather than simply encoding statistics of the pretraining corpus, remains an open empirical question, and the one this work addresses.

To this end, we compile a large cross-study cytometry corpus from ImmPort [1] spanning a variety of human immunological contexts, and adapt the variable-panel transformer architecture of Zhuang et al. [12] to pretraining at unprecedented cross-study scale. We evaluate the frozen encoder on a curated suite of 5 downstream tasks under a taxonomy of panel overlap conditions, as a principled examination of representation quality (more in Section 4), reporting scaling behaviour across encoder capacity and corpus size, and robustness to marker removal at test time. Our findings suggest that a single pretrained encoder can serve as a general-purpose feature extractor for clinical cytometry data, without requiring additional in-distribution pretraining.

We contribute:

1. **Infrastructure for large-scale cytometry modelling.** (1) A corpus of 89,324 quality-filtered samples from 112 studies spanning 525 distinct marker panels and 243 unique markers, the largest curated cross-study flow cytometry corpus for representation learning reported to date by study count. (2) A variable-panel transformer encoder pretrained on this corpus, the largest sample-level cytometry pretraining reported to date. We release the study and sample manifests, preprocessing scripts, code and model weights at `https://doi.org/10.5281/zenodo.23787478`.
2. **Evaluation framework for cross-panel transfer.** We propose a reusable evaluation framework structured around a taxonomy of panel overlap conditions — spanning seen (exact match or subset) and novel panels — that characterises transfer under panel heterogeneity. The framework includes a negative control task with no expected discriminative signal, frozen evaluation as a principled examination of representation quality for small clinical datasets, and diagnostic analyses covering scaling behaviour, robustness to marker removal, and few-shot performance. We further include linear probing of learned representations and a marker imputation test that assesses whether learned co-expression structure transfers to unseen data, readily adaptable to other variable-panel cytometry encoders.
3. **Empirical findings.** (1) The frozen encoder matches or exceeds in-distribution specialist encoders on 4 of 5 tasks, including all 3 novel-panel conditions and significantly exceeding them on HVTN; the negative control behaves as expected. (2) Scaling saturates under both encoder capacity and corpus random subsampling, despite continued improvements of the pretraining objective. (3) Representations degrade gracefully under marker removal and remain competitive in few-shot settings. (4) Self-supervised domain alignment via LoRA (Appendix C.1) yields inconsistent gains over frozen evaluation but outperforms the specialist on the single task where frozen fell behind.

2 Related Work

Machine learning for flow cytometry.

Prior machine learning work in cytometry has largely addressed single-study tasks: unsupervised cell population discovery, supervised cell type annotation, and sample-level phenotype prediction — without cross-panel transfer [13, 14, 15, 16, 17, 18]. Self-supervised pretraining for cytometry is more recent: SCARF applies contrastive learning within a single site [19]; FATE [10] introduces a variable-panel encoder but operates within a single clinical domain; GPCT [12] introduces the

Table 1: Comparison of pretrained cytometry encoders. *Pretrain samples*: for papers reporting cell counts only, marked N/R. *Pretrain/Eval overlap*: *same study* = pretraining and downstream are paired (train/test split applies to both); *no overlap* = downstream studies are disjoint from pretraining corpus. See footnotes for details.

Paper	Method	#Tasks	Panel Variability	Pretraining Method	Pretrain Samples	Pretrain/Eval Data Overlap
Lewis et al. [19]	MLP encoder + attn. pooling clf	1	—	Contrastive (SCARF)	1,820	Same study
FATE [10]	Transformer encoder (cell&sample[†])	1	cell encoder	MAE (sample)	756	No overlap
GPCT [12]	Transformer encoder (sample[†]) + attn. pooling clf	2	sparse cell embedding	Multi-task masked pred.	14,014	Same study
ImmuneFM [11]	Transformer encoder (cell[†]) + mean pooling clf	8	cell encoder	MAE (cell)	N/R (>100M cells)[‡]	Partial/ Unspecified[§]
This work	Transformer encoder (sample[†]) + attn. pooling clf	5	sparse cell embedding	Multi-task masked pred.	**89,324**	**No overlap**

[†] *Cell-level*: self-attention across markers within a cell. *Sample-level*: self-attention across cells within a sample.

[‡] No sample count reported; corpus described as "over 100 million single cells" [11].

[§] Four of eight downstream tasks use studies present in the pretraining corpus [11]. Downstream train/test split protocol not reported; it is unclear whether test samples from overlapping studies were withheld from pretraining.

variable-panel encoder, the sample classifier, and the multi-task pretraining objective we build on, but evaluates on two downstream datasets without studying pretraining scale. Concurrently, ImmuneFM [11] scales cytometry pretraining to 53 ImmPort studies using a cell-level encoder, demonstrating the value of scale with downstream fine-tuning. Our work is complementary: we focus on sample-level pretraining and systematically characterise frozen transfer under panel shift without fine-tuning, with strict separation between pretraining and downstream data.

3 Method

3.1 Problem setup: variable-panel flow cytometry

Let V denote a shared marker vocabulary of size $|V|$. In this study, a vocabulary with $|V| = 261$ is obtained by unifying and standardising marker names across the corpus prior to any quality filtering. A flow cytometry sample S consists of a set of cell marker expression vectors, featurised by a panel $M \subseteq V$. We represent each cell as a pair (x_i, a_i) where $x_i \in \mathbb{R}^{|V|}$ is a sparse marker expression vector (zero-filled for unmeasured markers) and $a_i \in \{0,1\}^{|V|}$ is a marker availability indicator with $a_{i,j} = 1$ iff marker $j \in M$. A learnable mask embedding μ_j replaces the zero entries of x_i for each unmeasured marker $j \notin M$ before the encoder, so that the model distinguishes structurally unmeasured from measured but unexpressed ones. The per-cell input to the encoder is the concatenation $c_i = [x_i; a_i] \in \mathbb{R}^{2|V|}$. The central modelling challenge is that M varies across samples: few studies measure exactly the same panel, so the encoder must produce representations that are comparable across heterogeneous inputs.

3.2 Encoder architecture

The encoder follows the transformer design of [12], which we summarise here; we refer the reader to that work for architectural derivations and motivation.

Cells within a sample are treated as an unordered set, with no positional encoding imposed. An input projection maps the $2|V|$-dimensional per-cell vector to a latent dimension d:

$$z_i = \mathrm{Proj}_{\mathrm{in}}(c_i) \;=\; \mathrm{LayerNorm}\big(\mathrm{Linear}\big(\mathrm{GELU}(\mathrm{Linear}(\mathrm{LayerNorm}(c_i)))\big)\big). \tag{1}$$

An L-layer pre-LN self-attention encoder [20, 21] then processes the cells, treating each latent cell vector z_i as a token. As the attention operates across cells within a sample, the encoder can capture inter-cell structure, e.g. subpopulation frequencies, co-expression patterns across cell types, that a

cell-level encoder alone cannot represent. A symmetric output projection produces per-cell latent representations $h \in \mathbb{R}^d$, which feed three prediction heads — one each for reconstruction, percentile, and density — each producing a predicted mean and a predicted variance (six outputs in total; see below). All heads are two-layer MLPs; variance heads apply Softplus to enforce positivity. We use $d = 512$, $L = 8$ encoder layers, 8 attention heads, feed-forward width $4d$, and dropout 0.1 for the main model configuration.

3.3 Pretraining objective: multi-task self-supervision

Following [12], we pretrain the encoder with a multi-task Gaussian negative log-likelihood (GNLL) objective that jointly predicts three cell-level quantities for every masked marker of every cell: (i) the reconstructed expression value y_{recon}, (ii) the per-sample rank percentile $y_{\text{pct}} \in [0, 1]$ of that cell's expression within its sample, and (iii) the estimated per-marker local density y_{dens} of the cell's expression . Each head predicts both a mean and a variance, and its loss is the Gaussian NLL under those predicted sufficient statistics:

$$\mathcal{L}^{(t)}_{\text{GNLL}}(\hat{\mu}, \hat{\sigma}^2, y) \;=\; \tfrac{1}{2}\log\hat{\sigma}^2 \;+\; \tfrac{(y-\hat{\mu})^2}{2\hat{\sigma}^2}, \qquad t \in \{\text{recon}, \text{pct}, \text{dens}\}. \tag{2}$$

Unlike point-wise reconstruction, percentile and density prediction require the model to represent where a cell sits in the sample-level distribution, a signal that cannot be read from any single cell in isolation and therefore forces cross-cell structure to emerge in the attention layers.

Masking. Each training step combines two masking passes with equal weight. The *uniform random* pass independently masks a fraction of (cell, marker) entries; the *marker-wise* pass masks entire markers sample-wide, so that the model must recover held-out marker expressions from the remaining panel. For each pass we compute per-head losses on the masked entries with a small residual weight on the unmasked entries:

$$\mathcal{L}^{(t)}_{\text{pass}} \;=\; (1-\beta)\,\mathcal{L}^{(t)}_{\text{GNLL}}\big|_{\text{masked}} \;+\; \beta\,\mathcal{L}^{(t)}_{\text{GNLL}}\big|_{\text{observed}}, \tag{3}$$

with $\beta = 0.1$. Note that we chose to normalise the weight on the masked entries accordingly so that both weights sum up to 1, a modification from the original approach [12]. The per-pass loss aggregates the three heads with task weights $(w_{\text{recon}}, w_{\text{pct}}, w_{\text{dens}})$:

$$\mathcal{L}_{\text{pass}} \;=\; w_{\text{recon}}\,\mathcal{L}^{(\text{recon})}_{\text{pass}} \;+\; w_{\text{pct}}\,\mathcal{L}^{(\text{pct})}_{\text{pass}} \;+\; w_{\text{dens}}\,\mathcal{L}^{(\text{dens})}_{\text{pass}}, \tag{4}$$

and the total step loss is $\mathcal{L} = \frac{1}{2}(\mathcal{L}_{\text{random}} + \mathcal{L}_{\text{entire}})$. Our models use $(w_{\text{recon}}, w_{\text{pct}}, w_{\text{dens}}) = (1, 1, 1)$.

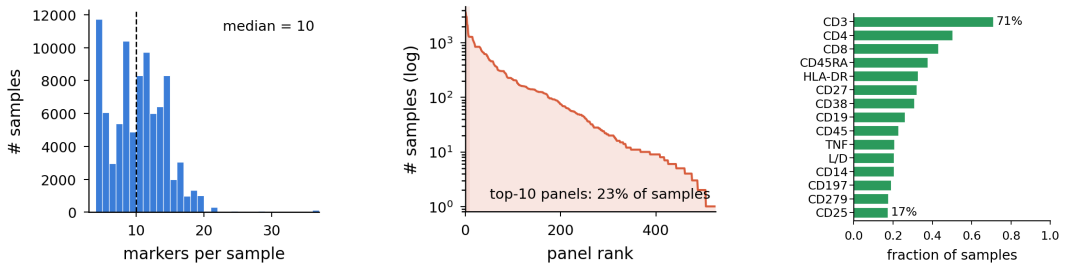
3.4 Pretraining corpus

We pretrain on public flow cytometry samples curated from the ImmPort repository [1]. The quality filtered corpus contains **89,324** samples from **112** studies ($\sim 4.5 \times 10^9$ cells, see Appendix B) , yielding 525 distinct marker panels and 243 unique markers. To our knowledge this is the largest curated cross-study flow cytometry pretraining corpus reported to date. The corpus spans a wide range of immunological contexts: infectious disease, vaccination, oncology, autoimmunity, and more. We release the study and sample manifests at `https://doi.org/10.5281/zenodo.23787478`.

The corpus is characterised by three properties that motivate our study (Figure 1): (a) *sparse observation*: median panel size is 10 markers, which corresponds to $> 95\%$ of the vocabulary structurally missing; (b) *heterogenous panels*: 525 distinct panels, with the top-10 covering only 23% of samples; (c) *skewed marker coverage*: CD3 appears in 71% of samples while specialised markers appear in $< 5\%$. Full corpus statistics are in Appendix Table 6.

4 Downstream evaluation protocol

Downstream tasks. We assemble **five** downstream tasks from four public clinical cytometry datasets, chosen to span distinct immunological contexts, acquisition protocols, panel designs, label granularities, and sample sizes. Table 2 summarises the tasks. Where a designated train/test split exists (HVTN) we honour it; otherwise we use leave-one-batch-out (LOBO) or batch-grouped stratified K-fold CV, grouping by acquisition batch. Batch-grouped evaluation is stricter than random



(a) markers / sample (b) samples / panel (log-rank) (c) top-15 marker coverage

Figure 1: Corpus heterogeneity. (a) Per-sample marker count. (b) Samples per panel (log-rank), showing a long tail of rare panels. (c) Marker frequency across the corpus.

CV because batch effects are the dominant source of technical variation in multi-site cytometry, so this protocol directly measures cross-batch generalisation. Tasks 1 and 2 share a dataset but target different biological axes (disease severity vs. sexual dimorphism), testing whether a single encoder captures multiple axes of immune variation. Task 5 is a **negative control**: the original study reported no discriminative signal [14], so near-chance performance is the expected pass condition. Dataset details are in Appendix B.

Table 2: Downstream evaluation tasks. LOBO = leave-one-batch-out CV; SGKF-5 = batch-grouped stratified 5-fold CV. Panel: overlap of the downstream panel with pretraining panels; *novel* = no pretraining panel contains all downstream markers (panel-zero-shot); *subset* = downstream markers are a subset of $\geq$1 pretraining panel; *exact* = panel appears in pretraining.

Task	Classes	N samples	Batches	Protocol	Panel	Source
1. COVID severity	6 (5 diseased + healthy)	119	6	LOBO	**novel**	[22]
2. COVID patient sex	2 (F + M)	134	6	LOBO	**novel**	[22]
3. HVTN stim antigen	2 (Gag. + Env.)	96	—	Split (0.5/0.5)	**novel**	[14]
4. B-neoplasm subtypes	9 (8 diseased + normal)	2251	364	SGKF-5	subset	[23]
5. HEU exposure	2 (HEU + UE)	308 (44 patients)	8	LOBO	exact	[14]

Evaluation procedure. For each task we compare two encoder regimes sharing an identical cross-attention-based classification head following Zhuang et al. [12] (implementation detailed in Appendix A), so any performance difference reflects the encoder quality. The **frozen pretrained encoder** uses ImmPort encoder weights without update; only the head is trained on labels. The **specialised (from-scratch) encoder** is randomly initialised, pretrained in-distribution on the training fold's unlabelled samples using the same self-supervised objective, then frozen and paired with the same head; its marker vocabulary is restricted to markers observed in the target dataset. Freezing is deliberate: it isolates representation quality in a regime where overfitting is frequent regardless of the adaptation strategy with small datasets, and preserves a single representational basis for comparison across tasks. On the three *novel*-panel tasks (Table 2), this constitutes a **panel-zero-shot** setting: the encoder must generalise to marker combinations never jointly observed during pretraining. Beyond the two encoder regimes, we also report a LoRA-based [24] domain alignment variant of the pretrained encoder in the Appendix C.1. We concatenate per-fold predictions for LOBO/SGKF tasks and compute AUC (OVR for multi-class) per seed, reporting mean $\pm$ std across seeds that control various sources of stochasticity in training. Paired bootstrap 95% confidence intervals are also computed on the AUC difference between the two encoders. As a non-deep-learning reference, we train a random forest on per-sample features built from per-marker summary statistics (mean, standard deviation and seven quantiles from 5% to 95% over each sample's cells), under the same data splits (Appendix C.3). Additional threshold-dependent metrics (F1, accuracy) are reported in the appendix.

Scaling and data efficiency. We examine how downstream transfer varies along three axes. **Pretraining corpus size**: we train encoders on random $\{10, 25, 50\}\%$ subsets of the full corpus, using uniform random subsampling without explicit dropping of entire study or panel. The training procedure is otherwise the same, with pretraining epochs scaled so that each encoder undertakes an equivalent number of optimisation steps, and each frozen checkpoint is evaluated on the downstream tasks following the procedure described above. In a follow-up investigation to probe corpus com-

position, we additionally pretrain on a volume-matched 10% corpus drawn from the fewest studies that together cover every downstream marker: a deliberately low-diversity counterpart to the random 10% subset (Appendix C.5). **Encoder capacity**: we train a larger encoder $5\times$ the parameter of the main model, with 12 encoder layers and a hidden size of 1024 (architecture details in Appendix A). We evaluate it on selected downstream tasks to test whether increased model capacity improves transfer. **Label budget**: for each task we restrict the classifier's labelled training set to K samples per class while keeping the encoder (and its self-supervised pretraining) unchanged, sweeping K over a task-dependent grid to perform few-shot analyses. Together these three axes test whether the encoder's utility scales with pretraining data and model capacity, and whether its representations remain useful under label scarcity.

Marker-panel stress test. Flow cytometry data is inherently partially observed: every sample measures only a subset of possible markers. We leverage this property to define a deployment-relevant robustness evaluation: clinical labs may use smaller or different panels than those seen during downstream training. At test time, we artificially restrict each sample's observed marker panel to a random size-k subset by masking the rest, sweeping k from 2 to the full per-task panel size, and measure downstream AUC using the already-trained classifier on masked input. Since the encoder's mechanism natively handles masked markers as unobserved, no architectural modification is needed. This tests whether the encoder's representations encode transferable biological structure beyond the specific markers observed — if the encoder merely memorises per-marker values, performance should collapse as k decreases rather than degrade gracefully. Masking is applied per sample over a grid of 10 mask seeds; per-$(k, \text{mask-seed})$ AUCs are averaged.

Intrinsic evaluation of pretrained representations. We evaluate the frozen encoder through two analyses that require no parameter updates. **1. Linear probing of encoder output.** Following the standard self-supervised evaluation protocol [25, 26, 4], we use a linear classification head (L2-regularised logistic regression with C selected via inner 3-fold CV) on mean-pooled cell-level encoder output, with the head trained to predict each task's class labels. We report macro-f1 and macro-averaged AUC under the same cross-validation scheme used for downstream classification. **2. Marker imputation test on the encoder.** To test whether the encoder has learned biologically meaningful co-expression structure that generalises to unseen data, a capability central to generalisation across studies with non-identical panels, we design a marker imputation test. Analogous to masked language model evaluation [3], we mask each observed marker as unobserved across all cells, and measure the frozen reconstruction head's ability to recover the masked marker expression. We report Pearson r between predicted and true values at cell and sample-mean level for all markers observed.

5 Experiments

Our experiments are organised around four questions. **(Q1)** Does the frozen pretrained encoder transfer across diverse clinical tasks and panel conditions (Section 5.1)? **(Q2)** How does transfer scale with corpus size, encoder capacity, and label budget (Section 5.2)? **(Q3)** How robust is the encoder to marker removal at test time (Section 5.3)? **(Q4)** What structure do the learned representations capture (Section 5.4)? We additionally report two analyses outside the evaluation protocol of Section 4: whether the pretraining objective itself saturates with corpus size (Appendix C.4), and how much each encoder's representation depends on acquisition batch (Appendix C.6).

5.1 Cross-study transfer performance

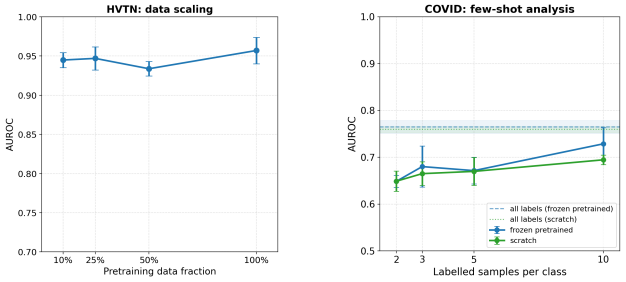
Main findings. Across the four signal-bearing tasks, the frozen pretrained encoder is competitive with or superior to in-distribution specialists. The clearest gain is on HVTN ($+4$ AUC points), a novel-panel task where the specialist has no structural advantage, and the one task with a CI that excludes zero. On COVID severity and COVID sex, the frozen encoder leads marginally ($+0.6$ and $+1.2$ points), both within seed variance. The fact that a single frozen encoder supports two distinct targets on the same cohort — severity and biological sex — differing only in the classifier head, indicates that the representation captures multiple axes of immune variation. Both methods score near chance (~ 0.48) on HEU, the negative control, confirming that the evaluation procedure does not lead to hallucinated signal where the source data provides none. On B-neoplasm, the specialist leads by 0.4 AUC points; this is the task with the most labelled data ($2{,}251$ samples) and a narrow

Table 3: Cross-study transfer: AUC (OVR for multi-class), seed-mean ± std with n_{seeds} in superscript. HEU is a negative control (near 0.5 = pass). Last row denotes 95% CI of Δ = pretrained − specialised AUC (paired bootstrap, 10,000 class-stratified resamples). Threshold-dependent metrics in Appendix Table 8; LoRA-SSFT in Table 7.

Method	COVID 6-class, LOBO	COVID sex bin, LOBO	HVTN bin, split	HEU bin, LOBO	B-neo 9-class, SGKF-5
RF (non-DL baseline)	$0.723^{(5)}_{\pm 0.002}$	$0.589^{(5)}_{\pm 0.005}$	$0.911^{(5)}_{\pm 0.009}$	$0.548^{(5)}_{\pm 0.007}$	$0.820^{(5)}_{\pm 0.002}$
Specialised, from scratch	$0.759^{(3)}_{\pm 0.007}$	$0.608^{(3)}_{\pm 0.026}$	$0.916^{(5)}_{\pm 0.023}$	$0.475^{(3)}_{\pm 0.053}$	$0.877^{(3)}_{\pm 0.003}$
Pretrained, frozen	$0.765^{(3)}_{\pm 0.014}$	$0.620^{(3)}_{\pm 0.013}$	$0.957^{(5)}_{\pm 0.017}$	$0.479^{(3)}_{\pm 0.082}$	$0.873^{(3)}_{\pm 0.0004}$
Δ, 95% CI	$[-0.042, 0.052]$	$[-0.057, 0.083]$	$[0.008, 0.084]$	$[-0.039, 0.049]$	$[-0.014, 0.005]$

Table 4: Encoder capacity scaling: medium (8L, d=512) vs. large (12L, d=1024, ∼5× parameters). AUC as seed-mean ± std; B-neoplasm large is a single seed as per seed variance has proven small.

	HVTN	B-neo	COVID sex
Med	$.957_{\pm.017}$	$.873_{\pm.0004}$	$.620_{\pm.013}$
Lrg	$.967_{\pm.007}$	.875	$.608_{\pm.014}$
Δ	+.010	+.002	−.012



(a) Corpus scaling (HVTN)

(b) Few-shot (COVID severity)

Figure 2: Scaling and data efficiency. (a) Downstream AUC vs. pretraining corpus fraction (HVTN; other tasks in Appendix C.9). (b) AUC vs. labelled samples per class (COVID severity; other tasks in Appendix C.8).

diagnostic panel well-matched to in-distribution pretraining. LoRA-based [24] domain alignment (Appendix Table 7) improves over frozen on B-neoplasm and COVID severity, but underperforms frozen on HVTN, the smallest dataset. This pattern is suggestive of a sample-size dependence, which calls for a systematic validation.

5.2 Scaling and data efficiency

Pretraining corpus size. Figure 2a shows downstream AUC as a function of the pretraining corpus fraction on HVTN (B-neoplasm and COVID severity in Appendix C.9). Performance is largely saturated by 10–25% of the corpus, demonstrating diminishing returns. Since subsets were drawn uniformly at random, even small fractions retain most of the corpus's study and panel diversity (the random 10% subset spans 109 of 112 studies, 402 distinct panels and 228 of 243 markers); the flatness of the scaling curve is consistent with representational quality being driven more by panel and study diversity than by sample volume. The volume-matched 10% low-diversity corpus drawn from only four studies (22 panels, 97 markers; Appendix C.5) lowers AUC on HVTN (0.925 vs. 0.945) and B-neoplasm (0.846 vs. 0.863) but raises it on COVID severity (0.777 vs. 0.757). One possible explanation is corpus composition in terms of panel: the dominant study (79% of samples) measures 19 of the 23 COVID panel markers but only 3 of the 8 HVTN markers, suggesting that the effect of diversity may depend on how much of the corpus closely matches the downstream panel.

Encoder capacity. Table 4 compares the main encoder with a ∼5× larger variant (12 layers, d=1024). The larger encoder produces marginal gains on HVTN (+1.0 AUC points, within seed variance) and B-neoplasm (+0.2 points) and slightly worse performance on COVID sex (−1.2 points, within seed variance), suggesting that encoder capacity is not the major bottleneck for downstream transfer.

Few-shot label efficiency. Figure 2b shows downstream AUC as a function of labelled samples per class on COVID severity (HVTN and B-neoplasm in Appendix C.8). The comparison is deliberately

conservative: the specialist encoder is pretrained on *all* unlabelled target-domain samples in each training fold, so it benefits from full in-distribution self-supervised signal regardless of label budget. Despite this advantage, the frozen cross-panel encoder maintains a competitive or higher operating curve across the K range, confirming that the cross-study representation carries enough task-relevant structure to substitute for in-distribution pretraining even at low label counts.

5.3 Marker-panel robustness

Figure 3 shows downstream AUC on B-neoplasm as markers are progressively removed at test time, with no retraining of the classifier (HVTN and COVID severity in Appendix C.10). Performance degrades gracefully: from 0.87 AUC at the full 7-marker panel to 0.73 at k=2, with the curve remaining above 0.83 at k=4 (just over half the panel). The pattern is consistent across tasks. The degradation is smooth, indicating that the encoder distributes task-relevant information across markers rather than relying on any single channel. The same holds at the representation level: dropping one observed marker leaves a cell's embedding at cosine similarity ≥ 0.98 to its full-panel embedding on every dataset, well above the encoder's own within-sample cell-to-cell similarity (Appendix C.7). The co-expression structure learned via the masked pretraining objective transfers to downstream classification under reduced marker panels, even on data unseen during pretraining.

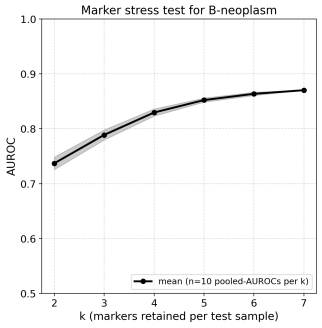
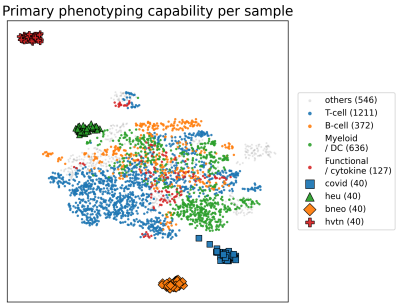
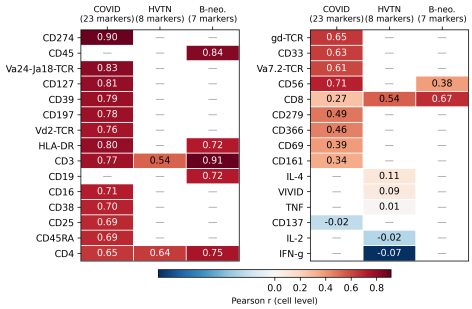


Figure 3: AUC vs. number of retained markers k, mean $\pm$ std over 10 mask seeds $\times$ classifier seeds (B-neoplasm).

5.4 Representation analysis



(a) UMAP of mean-pooled encoder embeddings: 2,892 pretraining samples and 40 subsampled downstream samples per dataset. All coloured by primary canonical lineage signature.



(b) Marker imputation test: per-cell Pearson r between imputed and true expression for each marker across three downstream datasets. Sample-mean r in Appendix C.11.

Figure 4: Representation analysis without supervision.

5.4.1 Linear probing

A logistic regression on the frozen, mean-pooled encoder embedding achieves AUC of 0.757 on COVID severity (6-class), 0.915 on HVTN (binary), and 0.828 on B-neoplasm (9-class). The modest gap relative to the full classification pipeline (Table 3) confirms that class-discriminative structure is largely present in the pooled representation itself, accessible to a simple linear model. Macro-F1 gaps are wider on the multi-class tasks (0.360 vs. 0.420 on COVID; 0.339 vs. 0.435 on B-neoplasm), indicating that the cross-attention classifier still adds value when resolving fine-grained class boundaries.

5.4.2 Embedding visualisation

We project mean-pooled encoder embeddings to 2D with UMAP [27] for 2,892 pretraining samples (up to 5 per (study, panel) bucket; 615 buckets) alongside 40 subsampled downstream samples from

each of the four evaluation datasets (Figure 4a). Each pretraining sample is coloured by the canonical lineage signature its panel best supports (T-cell, B-cell, Myeloid/DC, or Functional/cytokine; signature definitions in Appendix D).

Two patterns emerge from Figure 4a. First, the pretraining corpus organises by biology, not purely by study or panel identity: samples supporting the same lineage signature occupy locally coherent regions despite originating from different studies with hundreds of distinct panels. Panel size contributes to the embedding structure but does not disrupt the lineage-based structure (Appendix Fig. 10), which is expected as panels measuring fewer markers carry less biological information. Second, the four downstream datasets, which are not seen during pretraining, each form tight clusters rather than scattering as noise. HEU, having a primary myeloid / DC phenotyping panel with exact matches in the pretraining corpus, embeds close to the corresponding corpus region. The remaining three datasets, whose panels are either novel or a subset, land as coherent outliers at the periphery of the pretraining manifold. This is a visual confirmation that these tasks represent genuine distribution shift from the pretraining corpus, which makes the frozen transfer performance reported in Table 3 all the more notable.

5.4.3 Marker imputation test

Figure 4b shows per-cell Pearson r between imputed and true marker expression across the three signal-bearing downstream datasets (sample-mean r in Appendix C.11). The encoder accurately recovers core lineage and surface markers from co-expression context alone: CD3, CD4, and HLA-DR are imputed at $r > 0.7$ even from the 7-marker B-neoplasm panel ($r = 0.91, 0.75, 0.72$), with CD3 and CD4 exceeding their COVID-panel counterparts. In total, Pearson r exceeds 0.7 for 16 of 38 marker–dataset instances. Functional markers with sparser co-expression are harder to reconstruct (CD161: $r = 0.34$; CD69: $r = 0.39$ on COVID), and intracellular cytokines in the HVTN stimulation panel (IFN-γ, TNF, IL-2) are essentially unpredictable from surface markers ($r \approx 0$), consistent with the known biological independence of cytokine production from surface phenotype. Imputation quality tracks co-expression strength rather than panel size alone: CD56 achieves $r = 0.71$ on the 20–23-marker COVID panel but $r = 0.38$ on the 7-marker B-neoplasm panel, reflecting weaker co-expression support in the smaller context. These results confirm that the encoder has learned transferable marker co-expression structure, not panel-specific lookup tables.

6 Discussion and limitations

Frozen transfer as a practical default. The frozen result is stronger than it may appear: the specialist baseline is pretrained on all unlabelled target-domain samples, building a representation tailored to the target distribution, while the cross-study encoder must produce comparable representations across all tasks under significant panel and signal shift. That a single frozen encoder is competitive in this setting suggests frozen evaluation is not merely a methodological convenience but a viable deployment mode for clinical cytometry, where per-dataset pretraining is expensive or infeasible.

What scaling does not buy. Random corpus subsampling to 10% preserves most downstream performance, and a $5\times$ larger encoder yields no meaningful transfer gain despite achieving lower pretraining loss.[1] These two observations point in the same direction: the bottleneck for frozen transfer on the current task suite is not simply data volume or model capacity. The corpus scaling result is consistent with random subsampling preserving the study and panel diversity of the full corpus, and the low-diversity corpus (Section 5.2) suggests that the effect of diversity may be dependent on how much of the corpus closely matches the downstream panel. Quantifying this would require a designed sweep over corpus composition, which we leave for future work. The model-size result offers a complementary insight: it suggests that the pretraining objective can be optimised further without the improvement surfacing in downstream transfer, a form of diminishing returns at the representation level rather than the loss level.

Limitations and future work. The pretraining corpus is large for cytometry but small by ML standards ($89{,}324$ samples vs. millions in vision or language). The natural extension is to incorporate

[1] Mean loss over the last ten training epochs: -5.90 ± 0.007 (medium) vs. -6.24 ± 0.006 (large).

more data, such as those from FlowRepository [2]. All downstream tasks come from clinical immunology, and generalisation to other cytometry applications remains untested. The corpus also inherits the demographic composition of its source ImmPort cohorts. The tasks establish representational transfer, not clinical utility, which would require prospective cohorts with outcome data. We hold the GPCT objective fixed to isolate the effects of pretraining data and model size, leaving choices regarding hyperparameters (heads, β, cell subsampling, masking ratio) and alternative objectives to method development. Lastly, each downstream task uses a single panel and dataset. The stronger test is *downstream panel-zero-shot*: training a classifier on one or more panels and evaluating on different panels from different sites. Our results establish the precondition — a unified representation that applies to unseen panels. But whether a classifier built on those representations also generalises across panels remains an open question, and one we aim to explore in the future, alongside patient-level modelling that involves multiple cytometry samples of different panels.

7 Conclusion

We pretrained a variable-panel transformer encoder on 89,324 flow cytometry samples from 112 public studies and evaluated it, with weights frozen, on five clinical downstream tasks, three with marker panels never observed during pretraining. The frozen encoder matches or outperforms in-distribution specialists on 4 of 5 tasks and behaves as expected on a deliberately included negative control. Scaling analyses show saturation in both corpus subsampling and model capacity despite continued improvement in the pretraining objective, while few-shot experiments confirm competitive performance under restricted label budgets. The marker stress test demonstrates graceful degradation as markers are removed, and representation analyses confirm the encoder captures transferable biological structure — organising samples by lineage biology, and recovering marker co-expression patterns on unseen panels. Collectively, these results suggest that cross-panel, cross-study pretraining can produce a general-purpose feature extractor for clinical flow cytometry without task-specific retraining. Code, scripts and model weights are available at `https://doi.org/10.5281/zenodo.23787478`, with the implementation building on the GPCT codebase [12].

Acknowledgments and Disclosure of Funding

This research was undertaken with the assistance of resources from the National Computational Infrastructure (NCI Australia), an NCRIS enabled capability supported by the Australian Government.

T.D.A. and B.S.M. have been supported by the Jubilee Joint Fellowship scheme of the Australian National University.

The authors declare no competing interests.

References

[1] Sanchita Bhattacharya, Patrick Dunn, Cristel G Thomas, Barry Smith, Henry Schaefer, Jieming Chen, Zicheng Hu, Kelly A Zalocusky, Ravi D Shankar, Shai S Shen-Orr, et al. Immport, toward repurposing of open access immunological assay data for translational and clinical research. *Scientific data*, 5(1):1–9, 2018.

[2] Josef Spidlen, Karin Breuer, Chad Rosenberg, Nikesh Kotecha, and Ryan R Brinkman. Flowrepository: A resource of annotated flow cytometry datasets associated with peer-reviewed publications. *Cytometry Part A*, 81(9):727–731, 2012.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[5] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pages 491–507. Springer, 2020.

[6] Benjamin S Mashford, Timothy Hewitt, Maryam May, Zixin Zhuang, Akshat Jain, Koula EM Diamand, Fei-Ju Li, Kristy Kwong, Stuart H Read, Ainsley R Davies, et al. Voxelcoder: Classification of human cellular phenotypes via autoencoder batch alignment and hyperdimensional representation of cytometry data. *Patterns*, 2026.

[7] Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

[8] Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature methods*, 21(8):1481–1491, 2024.

[9] Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature methods*, 21(8):1470–1480, 2024.

[10] Lisa Weijler, Florian Kowarsch, Michael Reiter, Pedro Hermosilla, Margarita Maurer-Granofszky, and Michael Dworzak. Fate: Feature-agnostic transformer-based encoder for learning generalized embedding spaces in flow cytometry data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7956–7964, 2024.

[11] Sirui Ding, Sanchita Bhattacharya, and Atul J Butte. Immunefm: Pre-training foundation model from cytometry data for immunology research. *bioRxiv*, pages 2025–07, 2025.

[12] Zixin Zhuang, Benjamin S Mashford, Liang Zheng, and T Daniel Andrews. Towards a cytometry foundation model: Interpretable sample-level predictive modelling via pretrained transformers. *bioRxiv*, pages 2026–03, 2026.

[13] Sofie Van Gassen, Britt Callebaut, Mary J Van Helden, Bart N Lambrecht, Piet Demeester, Tom Dhaene, and Yvan Saeys. Flowsom: using self-organizing maps for visualization and interpretation of cytometry data. *Cytometry Part A*, 87(7):636–645, 2015.

[14] Nima Aghaeepour, Greg Finak, FlowCAP Consortium, Dream Consortium, Holger Hoos, Tim R Mosmann, Ryan Brinkman, Raphael Gottardo, and Richard H Scheuermann. Critical assessment of automated flow cytometry data analysis techniques. *Nature methods*, 10(3):228–238, 2013.

[15] Haidong Yi and Natalie Stanley. Cytoset: Predicting clinical outcomes via set-modeling of cytometry data. In *Proceedings of the 12th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 1–8, 2021.

[16] Zicheng Hu, Alice Tang, Jaiveer Singh, Sanchita Bhattacharya, and Atul J Butte. A robust and interpretable end-to-end deep learning model for cytometry data. *Proceedings of the National Academy of Sciences*, 117(35):21373–21380, 2020.

[17] Eirini Arvaniti and Manfred Claassen. Sensitive detection of rare disease-associated cell subsets via representation learning. *Nature communications*, 8(1):14825, 2017.

[18] Edgar E Robles, Ye Jin, Padhraic Smyth, Richard H Scheuermann, Jack D Bui, Huan-You Wang, Jean Oak, and Yu Qian. A cell-level discriminative neural network model for diagnosis of blood cancers. *Bioinformatics*, 39(10):btad585, 2023.

[19] Joshua E Lewis, Lee AD Cooper, David L Jaye, and Olga Pozdnyakova. Automated deep learning-based diagnosis and molecular characterization of acute myeloid leukemia using flow cytometry. *Modern Pathology*, 37(1):100373, 2024.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[21] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, pages 10524–10533. PMLR, 2020.

[22] David J Ahern, Zhichao Ai, Mark Ainsworth, Chris Allan, Alice Allcock, Brian Angus, M Azim Ansari, Carolina V Arancibia-Cárcamo, Dominik Aschenbrenner, Moustafa Attar, et al. A blood atlas of covid-19 defines hallmarks of disease severity and specificity. *Cell*, 185(5):916–938, 2022.

[23] Max Zhao, Nanditha Mallesh, Alexander Höllein, Richard Schabath, Claudia Haferlach, Torsten Haferlach, Franz Elsner, Hannes Lüling, Peter Krawitz, and Wolfgang Kern. Hematologist-level classification of mature b-cell neoplasm using deep learning on multiparameter flow cytometry data. *Cytometry Part A*, 97(10):1073–1080, 2020.

[24] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[25] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[26] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

[27] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[28] Holden T Maecker, J Philip McCoy, and Robert Nussenblatt. Standardizing immunophenotyping for the human immunology project. *Nature Reviews Immunology*, 12(3):191–200, 2012.

[29] Andrea Cossarizza, Hyun-Dong Chang, Andreas Radbruch, Andreas Acs, Dieter Adam, Sabine Adam-Klages, William W Agace, Nima Aghaeepour, Mübeccel Akdis, Matthieu Allez, et al. Guidelines for the use of flow cytometry and cell sorting in immunological studies. *European journal of immunology*, 49(10):1457–1973, 2019.

A Implementation details

Pretraining. We train the encoder with AdamW (weight decay 5×10^{-3} on transformer parameters) and a linear warmup–decay learning-rate schedule: linear warmup over the first 10% of steps, then linear decay from the peak rate (2×10^{-4}) to 10% of the peak. We used a held-in 5% validation split not for early stopping (patience set to n_epochs for full run), but for monitoring convergence at inference conditions (no neuron dropouts). Per-sample batch size is 64 and we subsample $4{,}000$ cells per sample on the fly. We use BFloat16 mixed precision training on a single NVIDIA H200 GPU for all experiments, except for the large model. A full 300-epoch run takes approximately 150 hours of wall-clock time on the $89{,}324$-sample corpus . Gradient clipping uses max-norm 40. Per-head losses use a GNLL numerical stability floor of 10^{-4} on predicted variances, clipped in $[10^{-4}, 10]$.

The large encoder variant (12 layers, 16 attention heads, d=1024; $\sim$5$\times$ the parameters of the main model) is trained with identical objective and hyperparameters on four H200 GPUs for 500 epochs.

The low-diversity 10% encoder (Appendix C.5) uses the architecture, objective and optimisation hyperparameters of the random 10% encoder; only the sample subset differs.

Each training step consists of two forward passes (random element masking and whole-channel masking), weighted equally at $\frac{1}{2}$ each. For the random element pass, the per-step masking rate is sampled from $\mathrm{Beta}(5.0, 16.0)$ and clamped to $[0.05, 0.4]$; for the whole-channel pass, the fraction of channels to mask is sampled from $\mathrm{Beta}(5.0, 15.0)$ (capped at 0.45, retaining at least 4 channels). A backward pass is performed after each forward pass, with the computation graph cleared after the first to save memory. The weight update is performed once both backward passes are complete. For both passes we use $\beta = 0.1$ as the residual weight on observed entries, and main-model task weights $(w_{\text{recon}}, w_{\text{pct}}, w_{\text{dens}}) = (1, 1, 1)$.

LoRA adaptation for downstream domain alignment. For the LoRA-SSFT regime we follow Hu et al. [24] and freeze all pretrained parameters while learning a low-rank additive update to every linear weight in the encoder,

$$W' \;=\; W + \tfrac{\alpha}{r}\, BA, \qquad A \in \mathbb{R}^{r \times d_{\text{in}}},\; B \in \mathbb{R}^{d_{\text{out}} \times r},$$

with A initialised via Kaiming-uniform and B initialised to zero, so the adapted model is identity-equivalent to the pretrained model at the start of fine-tuning. LoRA is applied to (i) the query, key, and value projections of every self-attention block as three independent rank-r adapters, (ii) the attention output projection W_o, and (iii) every `nn.Linear` in the feed-forward sub-blocks. In PyTorch's `nn.MultiheadAttention` the Q/K/V weights are stored as a fused tensor `in_proj_weight` $\in \mathbb{R}^{3d \times d}$; we realise independent per-projection adapters by stacking the three deltas $[\Delta W_q;\, \Delta W_k;\, \Delta W_v]$ before adding to the fused weight. Biases, LayerNorm parameters, and attention-weight dropout are left unmodified. Unless stated otherwise we use rank $r = 16$, scale $\alpha = 32$ (effective $\alpha/r = 2$), and LoRA-branch dropout $p = 0.1$ is applied to feed-forward linear layers, but not to the Q/K/V/output projection adapters. SSFT continues training with the same multi-task self-supervised objective used for pretraining (Section 3.3) on the training-fold samples (labels unused), using the same hyperparameter configuration as pretraining, for up to 800 epochs (matching downstream epochs). Only LoRA parameters receive gradients.

LoRA-SSFT vs. supervised LoRA-SFT. We deliberately choose a self-supervised LoRA fine-tuning step rather than supervised label-driven adaptation: with N in the dozens to few hundreds, a downstream classifier with sufficient expressivity easily memorises training labels while doing little to the representation. Keeping the pretraining objective on the training-fold samples aligns the encoder to the target cohort without letting label signal leak into the encoder, and we find this consistently outperforms supervised-LoRA variants in our preliminary studies.

Downstream classifier. The same classifier head is used across all three encoder regimes (from scratch, frozen, LoRA-SSFT). It consists of $L_d = 8$ cross-attention blocks (matching the encoder depth) with a single learnable prediction token attending over the per-cell encoder outputs, followed by a three-layer MLP prediction head; the encoder is frozen during classifier training in all three regimes. During training, whole-channel dropout is applied to the input at the same rate as neuron dropout (0.3), randomly masking entire markers across all cells in a sample; this encourages the decoder to base its decisions on robust cellular patterns rather than specific markers. We use AdamW,

learning rate 2×10^{-4}, batch size 16 (32 for B-neoplasm), dropout 0.3, and a fixed number of epochs per task chosen to ensure full convergence (250 - 1,000 depending on dataset size; see the last paragraph in Appendix A for the rationale). On B-neoplasm, the frozen encoder requires a moderately longer classifier schedule (500 vs. 300 epochs for scratch) to reach full convergence, consistent with its broader but less task-specific features needing more gradient steps to align with the 9-class target. Per forward pass we subsample up to 10,000 cells from a cached pool of maximum 100,000 cells per sample. For the from-scratch regime, the marker vocabulary is restricted to markers observed in the target dataset.

Data loading and caching. ImmPort FCS files are preprocessed into compact on-disk tensors indexed by sample ID. Per-sample cell caches (max of 60,000 cells for main encoder pretraining, 100,000 for downstream samples) are drawn once and sampled without replacement across epochs. Marker name normalisation uses the curated redirection table released with this work.

Hardware and compute budget. All DL experiments were run on NVIDIA H200 GPUs on a HPC cluster. Total compute for the experiments reported in this paper is approximately 2,300 GPU-hours, itemised in Table 5. Total project compute including preliminary and dropped experiments was approximately 2,600 GPU-hours. The RF based classical baseline runs on CPU only (12 cores, $\sim$11 hours).

Why train to convergence rather than early-stop. Each flow cytometry sample carries simultaneous biological signal along many axes: disease state, sex, age, ancestry, exposure history, technical batch. etc. Of those, only a few is labelled for each sample. Two samples that share a label on the categorical target of interest can therefore differ substantially on every other axis. Within a small clinical cohort (tens to a few hundred subjects) every sample is effectively unique in this latent description. There is consequently no well-defined joint distribution to match a held-out validation fold against the training fold; any validation draw reflects its own idiosyncratic mixture of unlabelled axes, and early stopping under such conditions selects a stopping point against val-fold idiosyncrasies rather than against generalisation. We therefore train every classifier in this paper for a fixed number of epochs chosen to ensure full convergence across every regime (scratch, frozen, LoRA-SSFT) and every task. This (a) uses all training labels rather than reserving samples for an early-stopping signal, (b) eliminates a hyperparameter (when to stop) whose tuning is unreliable in the small-N regime. The compute budget in Table 5 reflects this conservative choice; in deployment with larger cohorts or stronger held-out signal, a fraction of these epochs would suffice.

Table 5: Approximate GPU-hour budget for experiments in this paper.

Experiment	H200-hours
Pretraining (main model, 100% corpus, 300 epochs)	$\sim$150
Large-model pretraining (12L, d=1024, 4 GPUs, 500 epochs)	$\sim 200 \times 4 = 800$
Pretraining (scaling: 10%, 25%, 50% corpus and low-diversity 10%)	$\sim$600
LoRA-SSFT (5 tasks $\times$ folds $\times$ seeds)	$\sim$150
Downstream classifiers (5 tasks, scratch + frozen, full-N)	$\sim$200
Scaling + few-shot + large-model downstream evaluation	$\sim$200
Panel stress, linear probing, UMAP	$\sim$10
Downstream evaluation of additional encoders, third B-neoplasm seed	$\sim$165
Batch probing, marker-removal cosine analysis, objective-loss evaluation	$\sim$2
Total (reported experiments)	$\mathbf{\sim 2{,}300}$

B Dataset details

B.1 Pretraining data

All FCS files used for pretraining are publicly available through the ImmPort repository. The 112 study accessions and their samples can be found in the released corpus information.

Cell counts. The corpus is large not only in sample count but in total cells. Two cascading caps determine how many cells the encoder can access. First, at FCS-extraction time each sample is

capped at 100,000 cells (a storage constraint; 70.5% of files reach this cap, so the corpus total of $\sim 7.1 \times 10^9$ cells is a lower bound on the original FCS pool). Second, a per-sample cache cap retains 60,000 cells for the main encoder and 80,000 for the larger variant, yielding training-accessible pools of $\sim 4.5 \times 10^9$ and $\sim 5.8 \times 10^9$ cells respectively. Within each epoch a further subsample of 4,000 cells per sample is drawn on the fly, but every cached cell is reachable across the 300 training epochs. All cell-count estimates are bootstrapped from a uniform random draw of 1,000 files; 95% CIs are reported in Table 6.

Table 6: Pretraining corpus statistics. The corpus exhibits a long tail in both axes: most samples concentrate in a handful of small panels, but the panel-count distribution extends to rare 30+ marker configurations. †Cell counts estimated from n=1,000 uniformly random samples (bootstrap CI over the per-sample mean, scaled to 89,324). Each sample is capped at 100,000 cells on disk; the dataset loader further caps at 60,000 (medium encoder) or 80,000 (larger encoder) cells per sample for the training cache.

Samples		89,324
ImmPort studies		112
Marker vocabulary size $\|V\|$ (observed in corpus: 243)		261
Unique panels (distinct marker sets)		525
Samples per study (min / median / max)		3 / 357 / 9,068
Studies with ≥ 100 / $\geq 1{,}000$ samples		93 / 22
Samples per panel (min / median / max)		1 / 33 / 3,596
Top-10 panels' coverage of corpus		$\sim 23\%$
Per-sample marker count (min / p25 / median / p75 / max)		4 / 7 / 10 / 13 / 36
Per-sample on-disk cell count (mean / median;)†		79,788 / 100,000
Corpus-on-disk total (estimated, lower bound on FCS pool)†	$\sim 7.1 \times 10^9$	(95% CI $[6.93, 7.32] \times 10^9$)
Training-accessible total (medium encoder, cap 60,000)†	$\sim 4.5 \times 10^9$	(95% CI $[4.44, 4.62] \times 10^9$)
Training-accessible total (larger encoder, cap 80,000)†	$\sim 5.8 \times 10^9$	(95% CI $[5.70, 5.99] \times 10^9$)

B.2 Downstream data

COVID-19 severity (6-class LOBO). Peripheral-blood flow cytometry from the COMBAT dataset [22], distinguishing six immune conditions: `COVID_CRIT`, `COVID_MILD`, `COVID_SEV`, `Flu`, `Sepsis`, and `HV` (healthy). We exclude `COVID_LDN` (insufficient N=2) and `COVID_HCW` (ambiguous labelling). Evaluation is LOBO-CV over six acquisition batches. We use the study's T-cell panel only. Per-sample marker counts range from 20 to 23 as some FCS files leave some channel PNS fields empty, reducing the number of markers the preprocessing pipeline can resolve for those samples. Data is available via Zenodo as described in the original publication [22].

COVID patient sex classification (binary LOBO). Binary classification of biological sex (M/F) from the same COVID cohort and T-cell panel used for severity classification above. Sex labels are joined from the COMBAT clinical metadata release via the COMBATID linker; all 134 FCS files with a valid M/F annotation are included (75 M, 59 F), spanning the same 6 acquisition batches and using the same LOBO protocol. We note that sex is unevenly distributed across severity classes in this cohort (consistent with the known male skew in severe COVID/sepsis outcomes); LOBO-by-batch CV prevents subject leakage. Although sexual dimorphism in immune populations is well documented, age, demographics and disease status shape the same populations and all vary across these subjects; these competing sources of variation make sample-level sex classification hard in this cohort, and the classical baseline and every encoder variant we evaluate land between 0.59 and 0.63 AUC. Data is available via Zenodo as described in the original publication [22].

HVTN antigen stimulation (binary, designated split). Classifying *Gag*- vs. *Env*-stimulated samples from the FlowCAP II HVTN cohort [14], using the original train/test split (48/48). Unstimulated samples are excluded to match prior ML benchmarking practice. We note that is the tube-level stimulation-type classification task, not a responder/non-responder task. Data is available via FlowRepository (FR-FCM-ZZZV) as described in the original publication [14].

HEU vs. UE exposure (binary LOBO; negative control). Binary classification of HIV-exposed-uninfected (HEU) vs. unexposed (UE) infants. Each sample is (patient, dose) for 44 patients × 7

stimulation doses $= 308$ samples, grouped into 8 acquisition-date batches; we pool doses per fold and train one classifier per LOBO fold. The original study reported no discriminative signal between HEU and UE under this panel [14]. Data is available via FlowRepository (FR-FCM-ZZZU) as described in the original publication [14].

B-neoplasm 9-class subtyping (SGKF-5). Nine-class classification of B-cell neoplasm subtypes (`CLL`, `MBL`, `MCL`, `PL`, `LPL`, `MZL`, `FL`, `HCL`, `normal`) from the publicly released test cohort of Zhao et al. [23]. We follow the 9-class diagnostic system of the original study; samples carrying labels outside that system (`MM`, `HCLv`, `AML`) are excluded. A small number of files referenced by the original study were not retrievable through the FlowRepository API at the time of download and are omitted; the precise set is listed in the released manifest. Evaluation is batch-grouped stratified 5-fold CV, with batches defined by acquisition date. The source study uses three separate panels per sample; We evaluate on panel 3 of the original experiment, which the original authors report as the *least* informative of the three individual panels for subtype discrimination. Panel 3 is therefore the most demanding single-panel evaluation available from this study: the classifier must recover the full 9-class distinction from the study's hardest single tube, without access to the auxiliary information carried by panels 1 and 2. Data is available via FlowRepository (FR-FCM-Z2JW) as described in the original publication [23].

C Extended results

C.1 LoRA-SSFT panel alignment

Table 7 reports LoRA-SSFT AUC for all five tasks. Across the five tasks, LoRA-SSFT underperforms frozen where data is scarce (HVTN), but is a modest refinement where data appears more abundant (COVID severity, B-neoplasm); it is therefore reported here as an optional cohort-adaptation step.

Table 7: LoRA-SSFT downstream AUC, mean ± std across seeds (superscript: n_{seeds}). std for single seed is not reported. Compare against Table 3.

	COVID	**COVID sex**	**HVTN**	**HEU**	**B-neo**
LoRA-SSFT (multi-task)	$0.779^{(3)}_{\pm 0.014}$	$0.630^{(3)}_{\pm 0.029}$	$0.953^{(5)}_{\pm 0.021}$	$0.440^{(3)}_{\pm 0.057}$	$0.890^{(1)}$

C.2 Threshold-dependent metrics

Table 8 reports threshold-dependent metrics (accuracy and macro-F1) for the same configurations shown in Table 3. LoRA-SSFT results are included for completeness.

Table 8: Threshold-dependent downstream metrics, mean ± std across seeds. Accuracy and macro-F1 complement the AUC in Table 3; LoRA-SSFT from Table 7 is included here for a consolidated view.

Metric	**Method**	**COVID** 6-class	**COVID sex** bin.	**HVTN** bin.	**HEU** bin.	**B-neo** 9-class
Accuracy	Scratch	$0.445_{\pm 0.025}$	$0.587_{\pm 0.011}$	$0.892_{\pm 0.023}$	$0.494_{\pm 0.014}$	$0.834_{\pm 0.008}$
	Frozen	$0.457_{\pm 0.032}$	$0.600_{\pm 0.023}$	$0.917_{\pm 0.015}$	$0.486_{\pm 0.047}$	$0.840_{\pm 0.002}$
	LoRA-SSFT	$0.499_{\pm 0.038}$	$0.582_{\pm 0.053}$	$0.917_{\pm 0.021}$	$0.477_{\pm 0.046}$	0.845
Macro-F1	Scratch	$0.457_{\pm 0.026}$	$0.586_{\pm 0.011}$	$0.890_{\pm 0.024}$	$0.485_{\pm 0.012}$	$0.442_{\pm 0.020}$
	Frozen	$0.420_{\pm 0.039}$	$0.589_{\pm 0.022}$	$0.916_{\pm 0.015}$	$0.480_{\pm 0.045}$	$0.435_{\pm 0.009}$
	LoRA-SSFT	$0.492_{\pm 0.026}$	$0.573_{\pm 0.046}$	$0.916_{\pm 0.021}$	$0.470_{\pm 0.046}$	0.455

C.3 Classical baseline

As the downstream datasets were not released with gate definitions, we use per-marker summary statistics as sample-level features: for each observed marker, the mean, standard deviation and the 5, 10, 25, 50, 75, 90 and 95% quantiles over the sample's cells (up to 100,000), giving nine features per marker. Markers absent from a sample are median-imputed within each training fold. A random forest

(800 trees, class-balanced bootstrap subsamples) is tuned by inner stratified 3-fold CV on balanced accuracy over maximum depth $\{\text{none}, 10, 20\}$, features per split $\{\sqrt{\cdot}, \log_2, 0.5\}$ and minimum leaf size $\{1, 2, 4\}$, and evaluated with exactly the splits, sample filters and label exclusions of the deep pipeline. We report mean $\pm$ std over 5 forest seeds. On the negative control the baseline reaches 0.548 AUC, whose 95% stratified-bootstrap interval $[0.482, 0.616]$ contains chance.

C.4 Pretraining objective across corpus fractions

As each corpus-fraction encoder validates on its own subset during pretraining, per-run validation losses are not measured on comparable data. We therefore evaluate the 10, 25, 50 and 100% encoders on two common sets of 512 samples with identical cell sampling and the same masking draws (5 seeds), computing the pretraining loss exactly as in validation: a *held-in* set drawn from the random 10% slice, which is nested in every larger fraction and so seen by all encoders, and a *held-out* set drawn from the half of the corpus excluded from the 10, 25 and 50% subsets (Table 9). On held-in data the objective is flat, which naively resembles saturation; on held-out data it improves monotonically with corpus size. The objective's generalisation therefore keeps improving with data while downstream transfer does not.

Table 9: Pretraining objective loss (lower is better) on common evaluation sets, mean $\pm$ std over 5 masking seeds. The full-corpus held-out entry is parenthesised because that encoder trained on those samples.

Encoder	Held-in 10%	Held-out 50%
random 10%	$-5.954_{\pm 0.018}$	$-5.377_{\pm 0.054}$
random 25%	$-5.964_{\pm 0.010}$	$-5.732_{\pm 0.014}$
random 50%	$-5.953_{\pm 0.006}$	$-5.824_{\pm 0.005}$
full 100%	$-5.954_{\pm 0.007}$	$(-5.913_{\pm 0.007})$

C.5 Low-diversity 10% corpus

We build a markedly *less* diverse 10% corpus: the fewest studies whose markers together cover all 36 downstream markers (greedy set cover), subsampled to 8,932 samples to match the random 10% corpus. It draws on four studies, dominated by one (SDY2583: 7,023; SDY1335: 1,287; SDY1743: 464; SDY720: 158). Restricting the corpus lowers HVTN and B-neoplasm AUC but raises COVID severity (Table 10), whereas a uniform loss of diversity would be expected to hurt all three. A possible explanation is that the dominant study covers most of the COVID panel (19 of 23 markers) but little of the HVTN panel (3 of 8), so the effect of corpus diversity may depend on how much of the corpus closely matches the downstream panel.

Table 10: Low-diversity vs. random 10% pretraining (frozen evaluation, AUC, seed-mean $\pm$ std, n_{seeds} in superscript).

Pretraining corpus	Studies	Panels	Markers	HVTN	COVID severity	B-neo
low-diversity 10%	4	22	97	$0.925^{(3)}_{\pm 0.015}$	$0.777^{(3)}_{\pm 0.008}$	$0.846^{(1)}$
random 10%	109	402	228	$0.945^{(5)}_{\pm 0.010}$	$0.757^{(3)}_{\pm 0.010}$	$0.863^{(1)}$
full 100%	112	525	243	$0.957^{(5)}_{\pm 0.017}$	$0.765^{(3)}_{\pm 0.014}$	$0.873^{(3)}_{\pm 0.0004}$

C.6 Batch-effect probing

Every number in Table 3 is already an out-of-batch evaluation. To compare how the two encoders behave under that shift, we use the linear probe of Section 5.4.1 on mean-pooled sample embeddings (5,000 cells per sample vs. 10,000 in Section 5.4.1), varying only how its training data are chosen. *Batch held out* is the task's own batch-grouped CV. *Batch seen* is stratified CV in which the evaluated sample's batch appears in training through other subjects; each subject contributes one sample in the three tasks listed, so this isolates batch familiarity from same-subject similarity. For the specialist, each batch is evaluated with the fold encoder that held it out. Batch decodability is the balanced

accuracy of a logistic probe predicting batch from the embedding, under patient-grouped 5-fold CV over batches with at least five samples.

Table 11: Batch-effect probing. Penalty = batch-seen minus batch-held-out probe AUC. Batch-decoding chance is 0.167 for the COVID tasks (6 batches) and 0.008 for B-neoplasm (118 acquisition dates with $\geq$5 samples).

Task	Encoder	Batch decodable	AUC, batch seen	AUC, batch held out	Penalty
COVID severity	specialised	0.953	0.716	0.690	+0.026
	pretrained	0.913	0.745	0.746	−0.001
COVID sex	specialised	0.963	0.625	0.540	+0.085
	pretrained	0.918	0.653	0.605	+0.048
B-neoplasm	specialised	0.066	0.845	0.836	+0.009
	pretrained	0.045	0.826	0.821	+0.005

The pretrained encoder loses less when the evaluated batch is withheld, and batch is less decodable from it, on all three tasks (Table 11). Neither encoder is batch-invariant; the point is that a pretrained encoder which never saw these cohorts is no more batch-entangled than a specialist pretrained directly on them.

C.7 Marker removal in the representation

Section 5.3 measures marker removal through a trained classifier. Here we measure the shift in the representation itself, by inference only: markers are removed from a sample's panel (channel indicator set to unobserved, the structural-missing condition seen in pretraining), and the resulting embeddings are compared with the full-panel ones by cosine similarity. At the cell level (Table 12), we drop each observed marker in turn and use the similarity between different cells of the same sample as a reference scale. At the sample level (Table 13), we remove one marker or a random 25, 50 or 75% of the panel and compare mean-pooled sample embeddings; pooling averages out per-cell shifts, so similarities are higher than at the cell level. We evaluate all COVID and HVTN samples, 400 B-neoplasm, all HEU and 384 held-out pretraining-corpus samples, with $2{,}000$ cells per sample.

Table 12: Dropping one observed marker: cosine similarity between a cell's reduced-panel and full-panel embedding, against the within-sample cell-to-cell similarity under the full panel (mean over 3 cell-subsampling seeds).

	COVID	HVTN	B-neo	HEU	Corpus
one marker dropped (same cell)	0.990	0.984	0.982	0.992	0.994
different cells, same sample	0.864	0.944	0.882	0.887	0.874

Table 13: Removing a growing fraction of the panel: cosine similarity between reduced-panel and full-panel mean-pooled sample embeddings (mean over random subsets and 2 cell-subsampling seeds).

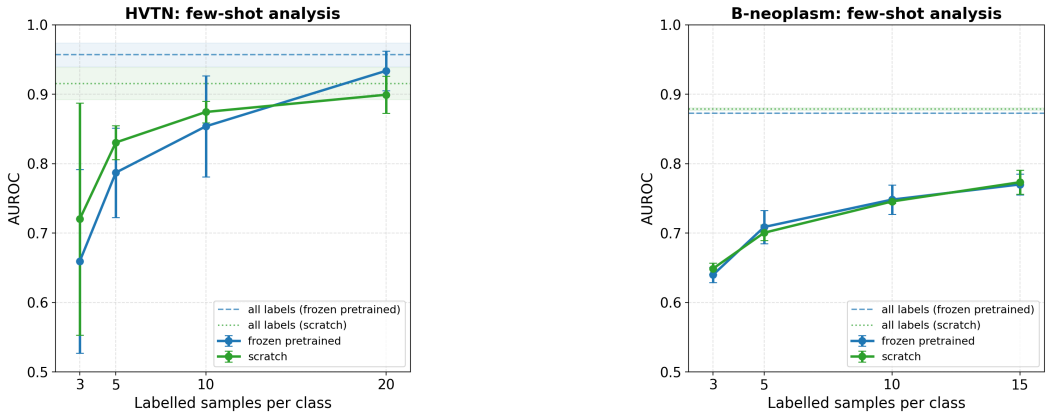
Panel removed	COVID	HVTN	B-neo	HEU	Corpus
one marker	0.995	0.990	0.991	0.995	0.995
25%	0.974	0.978	0.981	0.989	0.990
50%	0.950	0.953	0.956	0.972	0.976
75%	0.921	0.931	0.941	0.944	0.939

A single dropped marker moves a cell's embedding far less than the encoder's own spread between cells of one sample, and removing more of the panel degrades the sample representation smoothly: with 75% of the panel removed, sample-level similarity stays above 0.92 on every dataset.

C.8 Few-shot label-efficiency curves

Figure 5 shows few-shot label-efficiency curves for HVTN and B-neoplasm (COVID severity in main text, Figure 2b). The grids are: HVTN $K \in \{3, 5, 10, 20\}$ from 24/class; COVID severity

$K \in \{2, 3, 5, 10\}$ per fold; B-neoplasm $K \in \{3, 5, 10, 15\}$ per fold; three seeds per cell. The classifier epoch budget is increased for few-shot runs (train to convergence: 2,000 epochs for HVTN and COVID severity, 1,500 for B-neoplasm) because smaller training sets require more epochs to converge; all other hyperparameters are identical to the full-N setting (Appendix A). Self-supervised pretraining for the scratch baseline uses the full training fold regardless of K, so the few-shot restriction applies only to the classifier head's labelled data.



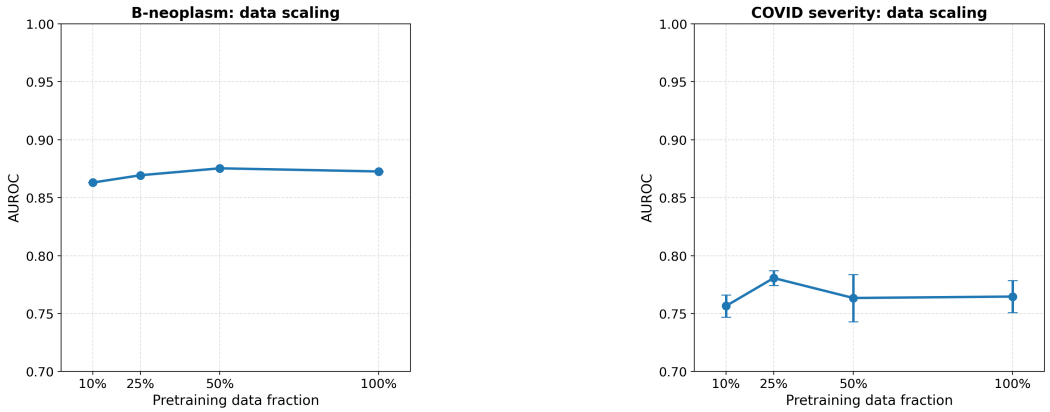
(a) HVTN (binary)

(b) B-neoplasm (9-class)

Figure 5: AUC vs. labelled samples per class for HVTN and B-neoplasm (frozen vs. scratch). COVID severity is in the main text (Figure 2b).

C.9 Corpus scaling curves

Figure 6 shows downstream AUC as a function of the pretraining corpus fraction for B-neoplasm and COVID severity (HVTN in main text, Figure 2a).



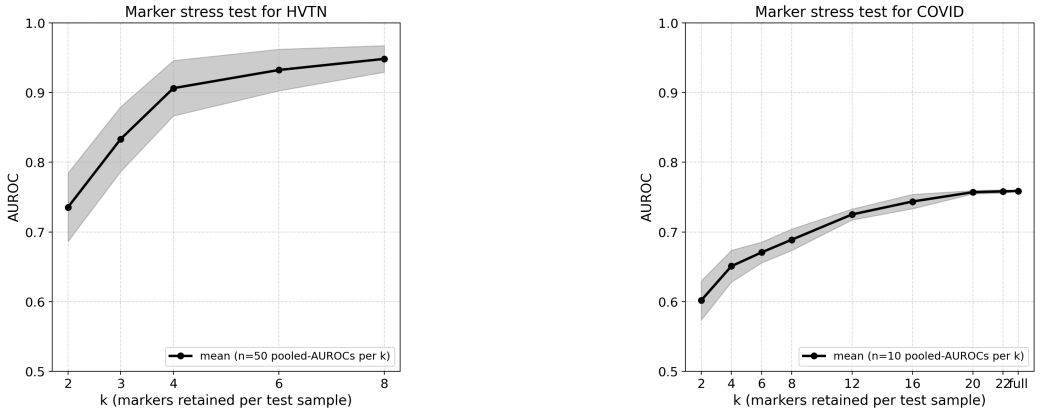
(a) B-neoplasm (9-class)

(b) COVID severity (6-class)

Figure 6: Downstream AUC vs. pretraining corpus fraction for B-neoplasm and COVID severity. HVTN is shown in the main text (Figure 2a). Overall, performance has a positive correlation with corpus size, but gains are marginal and unstable (e.g. covid performance spike at 25%)

C.10 Panel stress curves

Figure 7 shows marker-removal stress curves for HVTN and COVID severity (B-neoplasm in main text, Figure 3).



(a) HVTN (binary)

(b) COVID severity (6-class)

Figure 7: AUC vs. number of retained markers k for HVTN and COVID severity (frozen encoder, mean $\pm$ std over mask seeds $\times$ classifier seeds for HVTN, mean $\pm$ std over mask seeds for COVID).

C.11 Marker imputation: sample-mean

Figure 8 shows the sample-mean Pearson r between imputed and true marker expression for each marker-dataset pair, complementing the cell-level results in Figure 4b.

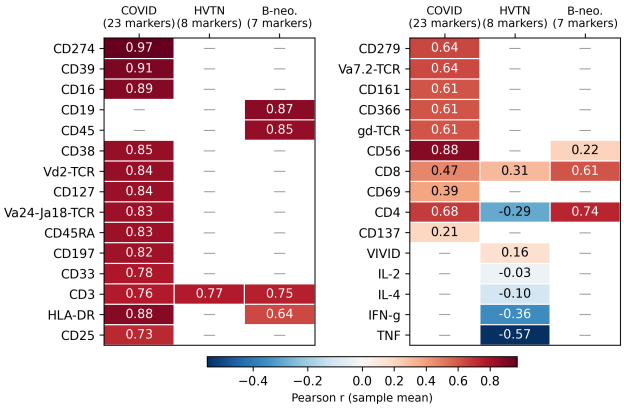


Figure 8: Sample-mean Pearson r between imputed and true marker expression across downstream datasets. Sample mean is taken over cells before computing the correlation

C.12 UMAP visualisations of corpus and downstream samples

Figures 9 and 10 show the UMAP embedding coloured by lineage signature and by panel size and study.

D Lineage signature panels for the representation analysis

Table 14 lists the markers used to score each sample's panel against the four lineage signatures introduced in Section 5.4. The signatures are pooled from the standardised Human Immunology Project (HIP) panels of Maecker et al. [28] and the consensus flow cytometry guidelines of Cossarizza et al. [29]. We deliberately use a coarse-grained, capability-level colouring rather than literal panel-to-panel replication: the goal is to visualise whether the encoder organises samples by which biology a panel can measure, not to recapitulate any particular published panel. A sample's panel supports a signature if its measured marker set contains at least k of that signature's markers; k scales with signature size. Markers absent from our vocabulary are omitted.

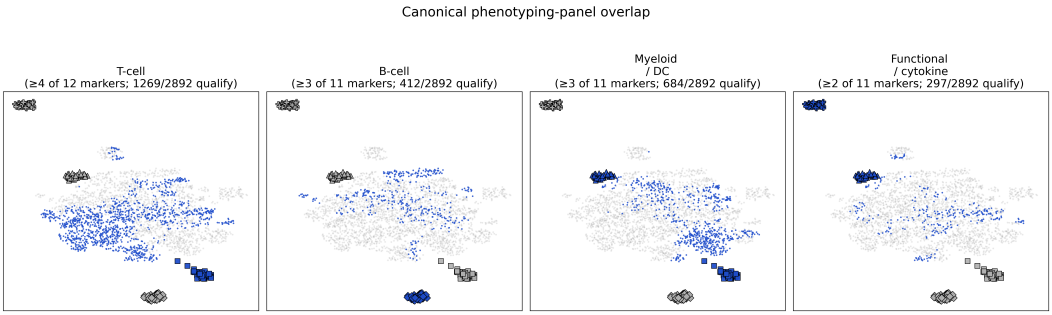


Figure 9: UMAP embedding coloured by lineage signature (Section D). Supplement to Figure 4a.

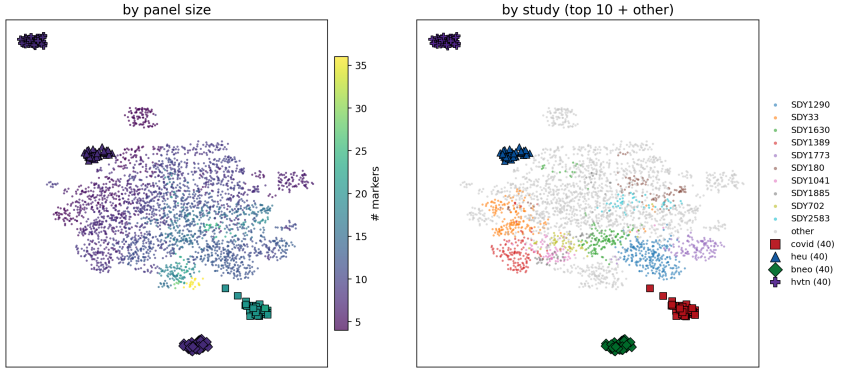


Figure 10: UMAP embedding coloured by panel size and study. Supplement to Figure 4a.

Table 14: Lineage signature panels used in Section 5.4. Markers are based on listed HIP panels of Maecker et al. [28] and selected additions from Cossarizza et al. [29]. A sample's panel *supports* a signature if it measures at least k of the listed markers.

Signature	Markers	k
T-cell phenotyping	CD3, CD4, CD8, CD45RA, CD45RO, CCR7 (CD197), CD27, CD28, CD127, CD25, CD62L, CD69	≥ 4
B-cell phenotyping	CD19, CD20, IgD, IgM, IgG, IgA, CD27, CD38, CD138, CD21, CD24	≥ 3
Myeloid / DC phenotyping	CD14, CD16, CD11b, CD11c, CD123, HLA-DR, CD15, CD66b, CD33, CD303, CD141	≥ 3
Functional / cytokine readout	IFN-γ, TNF, IL-2, IL-4, IL-6, IL-10, IL-17a, Granzyme B, Perforin, CD107a, Ki-67	≥ 2

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state three contributions, each supported in the experimental results.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We address the limitation of the paper in Discussion, under "Limitations and future work".

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper is an empirical study, contains no theorems or proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We described the implementation details in Methods and the Appendix. We described the evaluation protocols in the main text, and include dataset preprocessing details in the appendix. All data are publicly available. We release sample manifest and processing scripts. We release source code and training scripts.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: Pretraining data is publicly available from ImmPort; downstream datasets are publicly available from their respective repositories (cited in Section 4 and Appendix B). We release the sample manifest, marker synonym table, preprocessing scripts, evaluation code and Model weights at `https://doi.org/10.5281/zenodo.23787478`.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 4 specifies evaluation protocols and the seed-averaging procedure. Appendix A provides full hyperparameters for pretraining, LoRA adaptation, and the downstream classifier. Appendix B details dataset construction and filtering.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The main results table reports mean $\pm$ standard deviation across seeds, with the number of seeds shown in superscript, and paired bootstrap 95% confidence intervals (10,000 class-stratified resamples) on the pretrained–specialised difference. The source of variability (random seed controlling classifier initialisation and data sampling) is stated in Section 4. Supplementary tables follow the same convention; where a configuration has a single seed (e.g. B-neoplasm in certain appendix tables due to computational constraints), this is visible from the absence of a standard deviation.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix A states the GPU type (NVIDIA H200), and Appendix Table 5 itemises GPU-hours per experiment category.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: All data used is publicly available and de-identified. No new human data was collected.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: We do not foresee a direct path to negative societal impact. Flow cytometry data contains no patient-identifying information, unlike genomic sequencing. All data are from public, de-identified sources.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The released model is a cytometry feature extractor with no plausible dual-use risk. The data modality carries no patient-identifying information.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All datasets are cited with their original publications (Section 4, Appendix B). Pretraining data is obtained from ImmPort under the ImmPort User Agreement (permits unrestricted use and redistribution with attribution). The FlowCAP II datasets (HVTN, HEU) are from FlowRepository under its Terms of Use (no restrictions on use or redistribution; attribution required). The B-neoplasm dataset is from FlowRepository under the same terms. The COMBAT COVID dataset is released under CC-BY 4.0. The encoder architecture is credited in-text.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: We release corpus manifests, a marker synonym table, preprocessing scripts, evaluation code and model weights at `https://doi.org/10.5281/zenodo.23787478`. Corpus statistics are documented in Appendix B.

Guidelines:

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

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

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

Answer: [N/A]

Justification: This work does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: No new human subjects research was conducted. All data are from previously published, de-identified public datasets.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not a component of the core methods in this research.

    Guidelines:

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