Expert Systems With Applications 327 (2026) 132710

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Hierarchical supervision in DINOv2 training improves generalizability on white blood cell images

Manon Chossegros [a], Sophia Wagner [b,c], Christian Matek [c,d,e,f], Daniel Stockholm [g,h], Xavier Tannier [a,*], Carsten Marr [c,i,j,k,l]

[a] Sorbonne Université, Université Sorbonne Paris Nord, Inserm, Limics, France
[b] Department of Pathology, Mass General Brigham, Harvard Medical School, Boston, MA, USA
[c] Institute of AI for Health, Helmholtz Zentrum München, Germany
[d] Institute of Pathology, Universitätsklinikum Erlangen, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
[e] Comprehensive Cancer Center Erlangen-EMN, Universitätsklinikum Erlangen, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
[f] Bavarian Cancer Research Center, Augsburg, Germany
[g] PSL Research University, EPHE, France
[h] Sorbonne Université, Inserm, Centre de Recherche Saint-Antoine, CRSA, France
[i] Department of Physics, University of Munich, Munich, Germany
[j] Department of Medicine III, Ludwig-Maximilian-University Hospital, Munich, Germany
[k] Munich Center for Machine Learning (MCML), Munich, Germany
[l] DKTK, German Cancer Consortium, Munich, Germany

ARTICLE INFO

Keywords:
DINOv2
Hierarchical supervision
White blood cells
Foundation model
Latent space
Semi-supervised learning

ABSTRACT

The microscopic observation of blood cells is a crucial step in diagnosing pathologies such as leukemia. DINOv2 models have been employed to extract features from blood cell images, but they do not include biological knowledge, nor do they allow multi-granular labels. To enhance the representation of these cells, we propose leveraging a biologically informed hierarchy of white blood cell types. We train a DINOv2-based foundation model with a semi-supervised framework that uses hierarchical supervision. It enables using datasets with varying levels of label precision within a structure that represents the process of cell differentiation. To support multi-level label precision, we modify the original hierarchical loss function, allowing any hierarchy level to serve as a ground truth class. We evaluate our model on three external datasets, including an out-of-domain set of cervical cells. Our approach improves generalization of the model to new datasets, improving by 1 percentage point the balanced accuracy on the two blood cell external datasets, and by 2.5 percentage point the balanced accuracy on the out-of-domain dataset. In addition the proposed strategy better aligns the model's latent space with biological properties, leading to more acceptable misclassifications.

1. Introduction

The morphological characterization of white blood cells is important for diagnosing blood pathologies. It requires identifying and counting cells in the sample, a time-consuming process that is usually performed by an expert. The automation of white blood cell identification has been made possible by deep learning algorithms (Shahzad et al., 2024).

Self-supervised models are particularly useful in hematology because they do not need expert annotations to extract features from cell images. DINOv2 is the most commonly used architecture for self-supervised models; its latent space can capture essential information from cells without any labels (Oquab et al., 2304). The feature vectors contain meaningful characteristics that are usable for downstream tasks such as classification, clustering, and segmentation (Koch et al., 2024; Chen et al., 2024; Ding et al., 2411). In the context of hematocytology, Koch et al. (Koch et al., 2024) have trained a DINOv2 model on over 380,000 unlabeled white blood cells from 12 datasets. However, the absence of annotations can reduce the precision of the model in complex hematologic tasks, such as the detection of rare leukemic cells.

At the same time, the integration of labels coming from diverse data sources can be challenging, because labels are not necessarily harmonized between datasets; some sources provide only coarse classes, while

* Corresponding author.
E-mail address: xavier.tannier@sorbonne-universite.fr (X. Tannier).

https://doi.org/10.1016/j.eswa.2026.132710
Received 21 November 2025; Received in revised form 6 March 2026; Accepted 30 April 2026
Available online 7 May 2026

others provide more fine-grained classes. We call label precision the granularity of the label in the dataset. For example, Matek et al. (Matek et al., 2021) describe neutrophils as band neutrophils and segmented neutrophils, while Kouzehkanan et al. (Kouzehkanan, Saghari, Tavakoli, Rostami, Abaszadeh, Mirzadeh, Satlsar, Gheidishahran, Gorgi, & Mohammadi, 2022) refer to them simply as neutrophils. Thus, integrating different datasets requires harmonization of their respective labels, whether it is by discarding coarsely labeled images or by retaining only the most general labels, thereby losing information.

In order to include all labels that can be found in the literature, we propose to use hierarchical classification, where different degrees of label precision are given by different levels of the hierarchy. This is supported by the fact that white blood cell classes typically correspond to differentiation stages within the hematopoiesis tree, characterized by both their lineage and their degree of maturation (Rieger & Schroeder, 2012). For example, Diehl et al. (Diehl, Meehan, Bradford, Brush, Dahdul, Dougall, & He et al., 2016) develop a representation of blood cell types with a tree, based primarily on genomics and transcriptomics similarities. Several methods have been proposed to integrate hierarchical relationships into algorithmic training. One approach involves embedding labels into vector spaces where relative positions represent semantic relationships. For example, word encoders have been employed to capture relationships between labels and optimize compatibility between image embeddings and their corresponding labels (Akata et al., 2015; Frome et al., 2013). Alternatively, loss functions can be designed to ensure hierarchical consistency. Hierarchical cross-entropy loss defines an output class by the edges leading to it, an approach that has been applied to classification of cervical cell images (Bertinetto et al., 2020; Cai et al., 2024). Another strategy, Hierarchical SupConLoss, ensures that the attraction between samples and their anchors is proportional to their proximity within the class hierarchy (Zhang et al., 2022). Finally, hierarchical structures can be directly embedded into model architectures, such as by defining generalist and expert networks or adding output branches at different levels of classification (Ahmed et al., 2016; Zhu and Bain, 1709).

In this paper, a new framework is defined to introduce supervision into the training of a DINOv2-based model, by adding either a classification task or contrastive task when labels are available. Since classes are described at varying levels of label precision, they are represented in a hierarchy to preserve as much information as possible. The purpose of this pipeline is to combine the high generalizability of self-supervised models with improved interpretability through the use of a biologically informed hierarchy. The study makes the following contributions:

- A medically grounded hierarchy is designed to integrate all available labels from different data sources, and we implement it in a hierarchical loss (classification or contrastive task).
- Supervision is added in the training of a DINOv2 model for white blood cell images, which takes a hierarchical form.
- We modify the hierarchical loss function to include labels of different granularities.
- We test it on three different external datasets and show that their representation in the latent space is improved.

Code is available at https://github.com/mc2295/ dino_hier.

2. Material and methods

2.1. Datasets and classes

A description of the training datasets is presented in Fig. 1 and the testing datasets in Fig. 2. Following (Koch et al., 2024), 12 publicly available datasets are gathered for training, amounting to 380,000 images.

For testing, we use 3 external datasets. (i) The Acevedo dataset covers all possible cell types that are encountered in normal blood

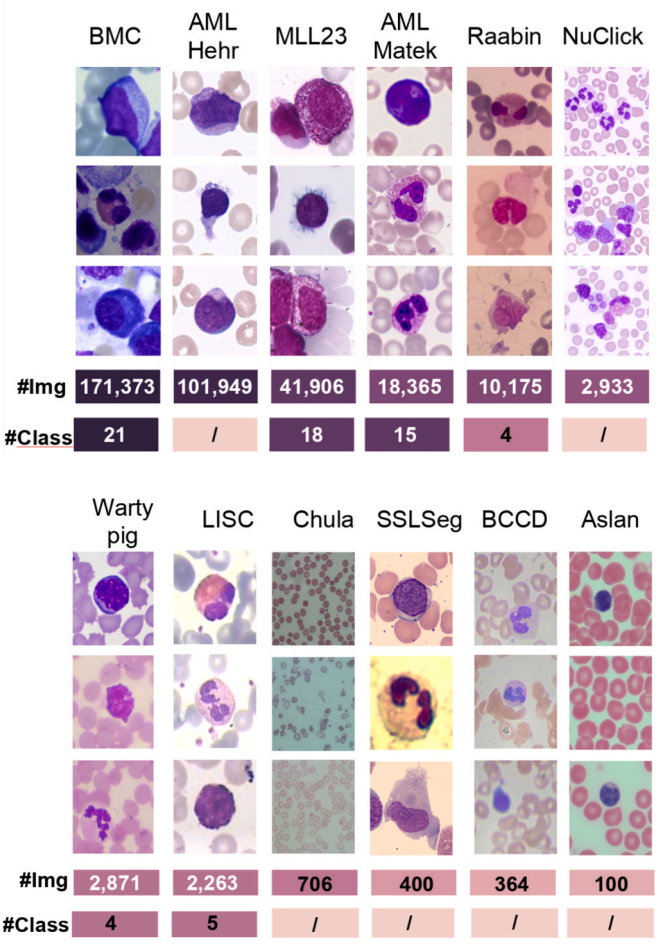


Fig. 1. Training datasets with the number of images and the number of class labels. AML Matek (Matek et al., 2019), LISC (Rezatofighi & Soltanian-Zadeh, 2011), Raabin (Kouzehkanan, Saghari, Tavakoli, Rostami, Abaszadeh, Mirzadeh, Satlsar, Gheidishahran, Gorgi, & Mohammadi, 2021), Bone Marrow Cytomorphology (Matek et al., 2021), MLL23 (Shetab Boushehri et al., 2025), NuClick (Koohbanani et al., 2020), Blood Cell Count and Detection (Sonar & Bhagat, 2015), Aslan (Aslan, 2020), AML Hehr (Hehr et al., 2023), SSLSeg (Zheng et al., 2018), Warty pig (J. Alipo-on, F. Escobar, J. Novia, M. Atienza, S. Mana-ay, M. Tan, N. AlDahoul, E. Yu, Dataset for machine learning-based classification of white blood cells of the juvenile visayan warty pig. 2022, 2022), Chula (Naruenatthanaset et al., 2012). Figure reproduced from (Koch et al., 2024) with permission from the authors.

samples, and is acquired with a CellaVision device. Images look noticeably different from the training images (Acevedo et al., 2020). (ii) The Bodyfluid dataset contains cells from body fluid samples rather than from blood smears so their morphology differs from training datasets. Additionally, even if most of the cells in this dataset are leucocytes, it also contains new classes, such as mitotic cells, mesothelial cells, etc. (Chomean et al., 2025) (iii) HiCervix is an out-of-domain dataset representing cervical cells that are morphologically completely different from leukocytic cells. Images are taken from three different hospitals with different cameras (Cai et al., 2024). For datasets with available cell-level labels, the detailed labels are represented in Appendix A.1. HiCervix dataset is chosen for out-of-domain performance because cervical cells share many similarities with blood cells: they are imaged and stained with similar procedure, their chromatin texture shows common patterns, and the dataset focuses on cell level characteristics. The purpose of this dataset is to show that some knowledge learnt from blood cell features can be transfered to other types of fluidic cells.

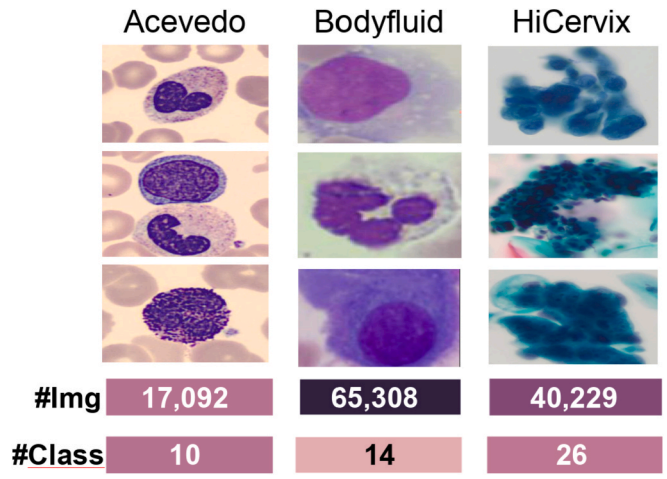


Fig. 2. Testing out-of-domain datasets: Acevedo (Acevedo et al., 2020), HiCervix (Cai et al., 2024), Bodyfluid (Chomean et al., 2025).

2.1.1. Hierarchical representation of the classes

A hierarchy is designed with clinical experts to align with the biological characteristics of white blood cells and to include all available labels. The output is shown in Fig. 3. In this hierarchy, white blood cells are organized as follows: Lineage > Maturity > Cell types > Additional morphological characteristics. This hierarchy was validated by hematological experts and largely aligns with previous work by Matek et al. (2019) (Matek et al., 2019), as well as with the Cell Ontology mapping (Diehl, Meehan, Bradford, Brush, Dahdul, Dougall, He, & The cell ontology, , 2016). We chose to split first by lineage and then by maturity, as this configuration is biologically valid and better aligns with the visual similarity between cells. As noted by Bertinetto et al. (Bertinetto et al., 2020), the hierarchy should not deviate significantly from visual similarity. The graphical representation of the hierarchy is called a tree, where the extreme nodes are called leaves, the intermediate nodes are internal nodes, and the starting node is the root. In this tree, child classes inherit from their parent classes, meaning that lower levels must be subgroups of higher levels.

2.2. DINOv2: a self-supervised trained model

DINOv2 is traditionally trained using self-supervised learning. We choose this model because of its remarkable performances for feature extraction, especially on out-of-domain datasets (Oquab et al., 2304). The goal is to learn meaningful representations of the images without requiring labels. It combines the DINO teacher-student framework (Caron et al., 2021) with masked image modeling inspired by iBOT (Zhou et al., 2021). The self-supervised training of DINOv2 is shown in Fig. 4.

In the training process, different augmented versions – called views – of an image are given to the network. They are global crop and local crop augmentations, depending on the size of the crop. In our case, only two global crops are used, because white blood cell local crops tend to remove important information, degrading the training, as described previously (Koch et al., 2024).

The training of DINOv2 is based on two components to extract both global and local features: first, the [CLS] token from the student is trained to match the [CLS] token from the teacher, after both pass through the DINO head. Second, some patches from the student's patch tokens are masked. The iBOT head is trained to reconstruct them using the corresponding unmasked teacher patches.

The model is trained using knowledge distillation; weights are computed from student to teacher using exponential moving average. DINOv2 also introduces the Sinkhorn-Knopp centering technique to balance and normalize the learned features, and KoLeo regularization to encourage uniform distribution of representations. More details are given in Caron et al. (Caron et al., 2021).

2.3. Model extensions

We propose two modifications in the traditional DINOv2 training pipeline. First, we introduce supervision in addition to self-supervised learning. Second, we make this supervision hierarchical, meaning the classes of the cells are represented in the form of a tree instead of a flat vector.

2.3.1. Supervision into DINOv2 training

To add supervision into DINOv2 training, we successively train the model with a classification task and with a supervised contrastive task. Each loss term is assigned an equal weight of 1.0, so the final loss is:

$$L_{semi-supervised} = L_{DINO} + Li_{BOT} + L_{supervised} \quad (1)$$

Supervised classification. For the classification, a supervised MLP head is placed on top of the model. It takes as an input the [CLS] token from the teacher embedding and predicts the image class. The last layer of the MLP head is used to compute cross-entropy. The configuration is illustrated in Fig. 5.

Cross-entropy (CE) quantifies the difference between the predicted probability vector $\hat{y}$ and the True label vector y. Its formula is given by:

$$\mathrm{CE}(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log \widehat{y_i} \quad (2)$$

We notice in this formula that the vector y_i is 1 on the actual class and 0 elsewhere, meaning that only the prediction of the actual class

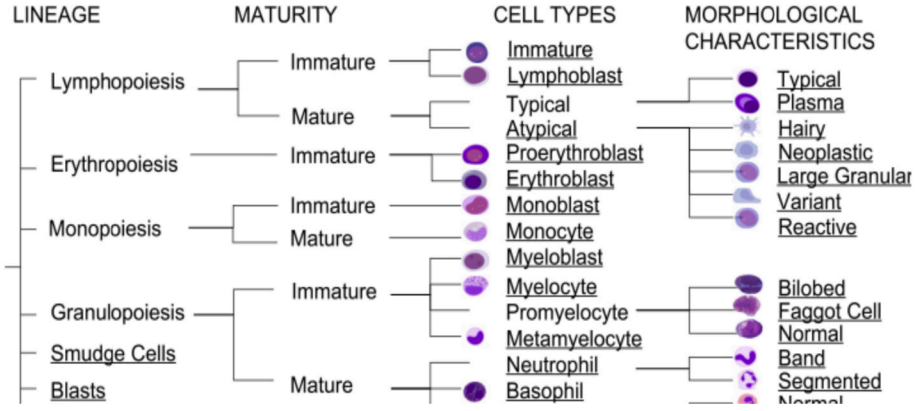


Fig. 3. Blood cell hierarchy. Classes are split by lineage, maturity, cell type and morphological chracteristics. White blood cell labels available in the datasets are underlined. Images are released under the GFDL-self license and are freely distributable.

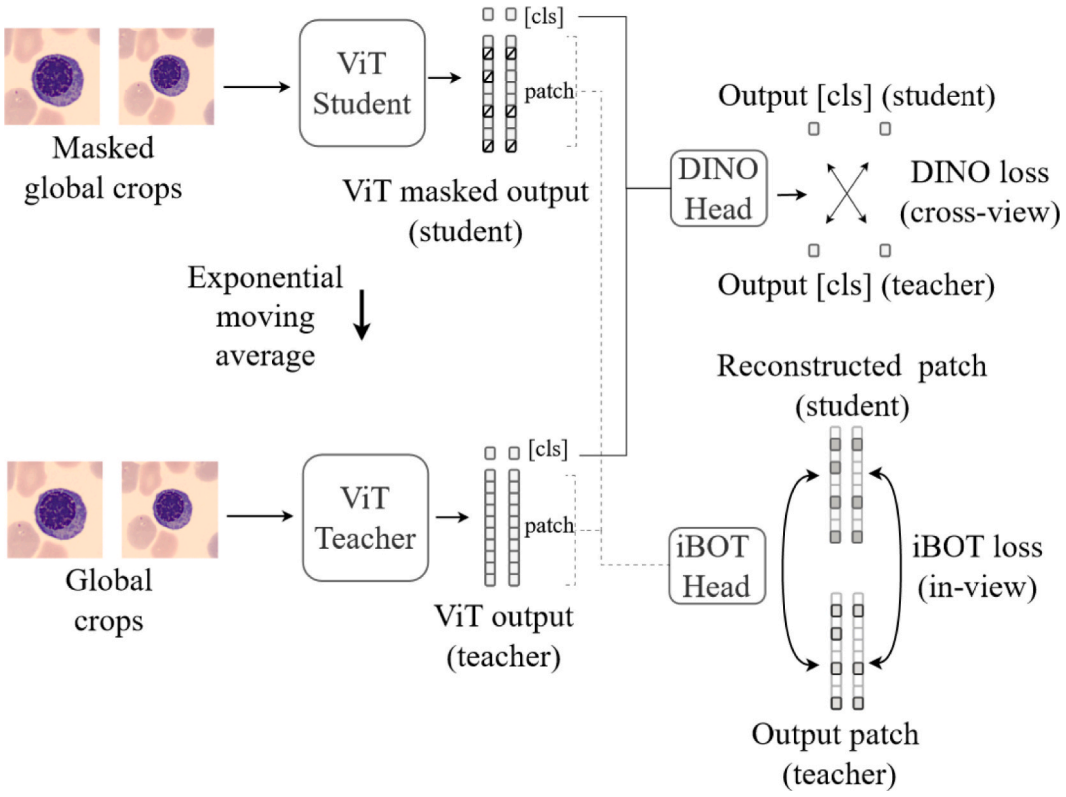


Fig. 4. DINOv2 training for white blood cell images wherenly global crops are used. The model is trained for self-supervised task with DINO loss and iBOT loss. The DINO loss matches the [CLS] tokens from *cross-view* pairs between student and teacher outputs, and the iBOT loss matches the patch tokens from *in to view* pairs between student's masked reconstructed patches and teacher's unmasked patches.

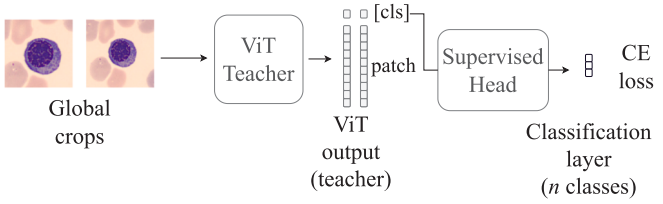


Fig. 5. Supervision is introduced into DINOv2 with a classification task. An MLP head is added after the [CLS] of teacher embedding. The last layer has n class nodes, and cross entropy is computed.

contributes to the loss.

Supervised contrastive learning. Another form of supervision can be introduced through supervised contrastive learning. Contrastive objective compares samples rather than treating them independently as cross entropy does. The model is trained to bring the representations of images that share the same label closer together. In this study, supervised contrastive loss (SupCon) is directly computed on the [CLS] token from the teacher embedding. Fig. 6 illustrates the computation of this loss.

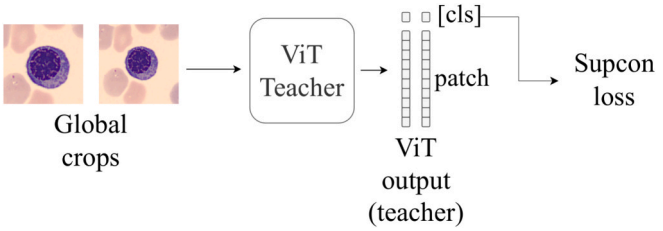


Fig. 6. Supervision is introduced into DINOv2 with supervised contrastive task. SupCon loss is computed directly on the [CLS] part of teacher embedding.

SupCon aims to maximize the similarity between images sharing the same label and to minimize the similarity between images of different labels (Khosla et al., 2020), it is calculated by:

$$L^{\text{sup}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P} \log \frac{\exp\left(\frac{z_i \cdot z_p}{\tau}\right)}{\sum_{a \in A(i)} \exp\left(\frac{z_i \cdot z_a}{\tau}\right)} \tag{3}$$

where $i \in I$ are the views in the batch, z_i their representation by the model, and τ is a temperature parameter. For each anchor view i, $P(i)$ are the positives, and $A(i)$ are the negatives.

2.3.2. Introduction of hierarchy in the supervised training

This part introduces two new losses, HierCE and HierSupCon, which are modified versions of the previously defined cross entropy and SupCon, adapted to include the hierarchical structure of the labels.

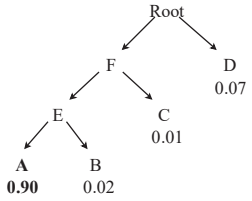
Hierarchical-cross entropy (HierCE). In equation 2, the computation of cross entropy does not take into consideration the output of other nodes, that are not the actual class. In hierarchical cross-entropy, instead of considering only the output of the actual node, other nodes also contribute to the loss, with weights depending on how close these classes are to the actual node. The mathematical details are provided in (Bertinetto et al., 2020).

Let us consider an example tree, shown in Fig. 7a, with a predicted value for each node.

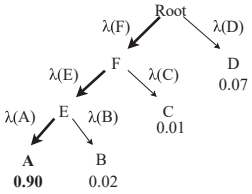
For the predicted vector $\hat{y}_i$, the probability of the **edge** between A and E is called $p_i\,(A|E)$, and it is computed as follows:

$$p_i(A|E) = \frac{\sum_{x \in \text{leaves under A}} \hat{y}_i(x)}{\sum_{z \in \text{leaves under E}} \hat{y}_i(z)} = \frac{0.90}{0.90 + 0.02} \tag{4}$$

The hierarchical cross-entropy is calculated as the sum of the logarithms of the probabilities of the edges leading to A, each weighted



(a) Example of hierarchical tree with a predicted value for each possible class.The actual class is A.



(b) The edges used for the computation of hierarchical cross entropy are in bold,leading to the actual class A. Each edge is weighted by a factor λ corresponding to the height of the edge in the tree.

Fig. 7. Computation of hierarchical cross entropy from the predictions of each class ordered in a tree.

according to the height of the edge in the hierarchy. These edges are shown in bold in Fig. 7b.

The final formula of hierarchical cross-entropy for a probability distribution p with classes C is given by:

$$HierCE(p, C) = -\sum_{l=0}^{h-1} \lambda(C^l) \log\left(p(C^l | C^{l+1})\right) \tag{5}$$

where $C^0,\ldots C^{h-1}$ are the classes on the path that connects the actual class to the root and h is the depth of the hierarchy. The value of $\lambda(C^l)$ is calculated with the following formula.

$$\lambda(C^l) = \exp\left(-\alpha h(C^l)\right) \tag{6}$$

here, $h(C^l)$ is the height of the class C^l and α is a hyperparameter. In this study, we set α to 0.5, which gives the best compromise between low and high level penalization (Bertinetto et al., 2020).

Modification of the HierCE for internal nodes. HierCE, as defined in Eq. (5), is calculated from the formula in (Bertinetto et al., 2020), but it does not allow non-leaf nodes to be predicted by the model. However, the purpose of the hierarchy in this study is to include images whose labels may come from higher levels in the hierarchy. For example, cells labeled as neutrophils should be included in training, even if it is not specified whether they are band neutrophils or segmented neutrophils. Therefore, we make a modification to the HierCE formula: in the case of a non-leaf class, the loss is calculated as the sum of its descendant leaf nodes. This is motivated by the fact that the probability of having class E can be written as the sum of the probabilities of having one of its child classes, A and B. The pseudo-code of the modified HierCE algorithm is given below, with our modifications highlighted in red.

Pseudo code of modified HierCE algorithm.

Modified HierCE:_init_
Input:
- Tree hierarchy T
- Leaf class list $\mathscr{L} = [l_1, ..., l_N]$
- Internal class list $\mathscr{I} = [i_1, ..., i_N]$
- Edge weight tree W (same structure as T)

Output:
- *onehot_num* $\in \mathbb{R}^{(N_L+N_I)\times N_L \times D}$
- *onehot_den* $\in \mathbb{R}^{(N_L+N_I)\times N_L \times D}$
- *weights* $\in \mathbb{R}^{(N_L+N_I)\times D}$ With D maximum depth

onehot_num = torch.zeros($(N_L + N_I) \times N_L \times D$)

(*continued on next column*)

(*continued*)

Pseudo code of modified HierCE algorithm.

onehot_den = torch.zeros($(N_L + N_I) \times N_L \times D$)
for each leaf class l ∈ $\mathscr{L}$ do
  for depth *d*, edge *e* in *edges_from_leaf*[*l*] do
  # we move along the path from leaf class to root
    # Numerator at depth *d*: all leaf nodes under the edge *e* are 1
    for each leaf *k* ∈ *leaves_under_edge*[*e*] do
      onehot_num[l, *k*, *d*] ← 1
    end for
    # Denominator at depth d: all leaf nodes under edge *e* and siblings of edge *e* are 1.
    for each leaf *k* ∈ *leaves_under_edge*[*e*] do
    onehot_num[l, *k*, *d* + 1] ← 1
    end for
  # Weight for this edge
    weights[l, *d*] ← W[*e*]
  end for
for each internal class *i* ∈ $\mathscr{I}$ do
  for each leaf *l* ∈ *descendant_leaves*[*i*] do
    onehot_num[i] ← min(*onehot_num*[*i*] + *onehot_num*[l], 1)
    onehot_den[i] ← min(*onehot_den*[*i*] + *onehot_den*[l], 1)
  end for
  # Aggregate weights (mean over descendants)
  for depth *d* do
    weights[*i*, *d*] ← mean(weights[*l*, *d*] for *l* ∈ *descendant_leaves*)
  end for
end for

Modified HierCE: forward

Input:
- Leaf probability vector p ∈ $\mathbb{R}^{B\times N_L}$
- Target class y ∈ $\mathscr{L} \cup \mathscr{I}$

Output: scalar loss
p ← unsqueeze(*p*, dim = 1)
num ← matmul(*p*, *onehot_num*[y])
den ← matmul(*p*, *onehot_den*[y])
for each valid depth *d* do
  loss[*d*] ← -log(*num*[*d*] / *den*[*d*])
end for
loss ← sum_d(*weights*[*y*, *d*] · loss[*d*])
return mean(*loss*)

An example is presented in Fig. 8; if the actual class is E, the edges under E serve for the computation of the loss.

The final computation of HierCE for intern node is written as follows:

$$HierCE(p, C_{intern}) = -\sum_{C\in \text{leaves}(C_{\text{intern}})} \sum_{l=0}^{h_{\text{intern}}-1} \lambda(C^l) \log\left(p(C^l | C^{l+1})\right)$$

$$-\sum_{l=h_{\text{intern}}}^{h-1} \lambda(C^l) \log\left(p({C_{\text{intern}}}^l | {C_{\text{intern}}}^{l+1})\right) \tag{7}$$

Hierarchical supervised contrastive loss (HierSupCon) is inspired by SupCon, but additionally integrates the hierarchical structure in the labels, as described by (Zhang et al., 2022) (see Fig. 9).

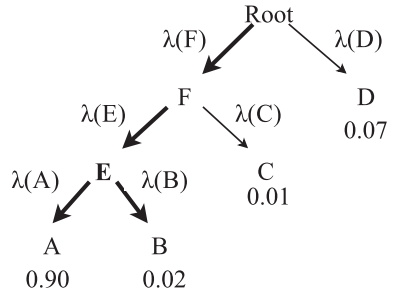
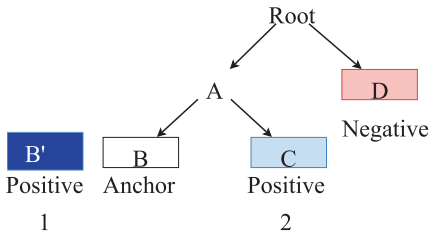
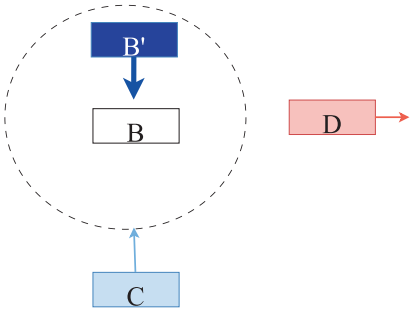


Fig. 8. Modified loss for the internal node E. The edges under E serve for the computation of the loss of the actual class E.



(a) Presentation of negative and positive samples with hierarchically organizedlabels: B is the anchor, B' and C are positive samples, D is a negative sample.



(b) The action of the loss in the embedding space is depicted. B' is stronglyattracted to the anchor because they share the same label. C is moderately attracted to the anchor, and cannot go closer than B', because they share only the same parent (A). D is negative so it is repelled.

Fig. 9. Description of HierSupCon loss.

$$L^{\text{HierSupCon}} = \sum_{l\in L}\frac{1}{|L|}\sum_{i\in I}\frac{-\lambda_l}{|P(i)|}\sum_{p_l\in P_l}\max\left(L^{\text{pair}}\left(i,p_l^i\right), L^{\text{pair}}_{\max}(l-1)\right)$$

With l being the level in the hierarchy, λ_l a controlling parameter that applies a fixed penalty for each level in the hierarchy, and P_l the set of positive images for anchor image i. We set $\lambda_l = \frac{1}{l}$. L^{pair} is the contrastive loss for a pair with the anchor image.

This has two consequences on the loss: first, positive samples can also be images with different labels, as long as they share at least one common ancestor label with the anchor, other than the root. The level of attraction of positives depends on how close their class is to the anchor in the tree. Second, a constraint is imposed to ensure that the distances between samples respect the relative distances in the tree. In other words, images sharing the same labels should be closer than images sharing only the same parent, and so on. Finally, these two terms are combined to give HierSupCon loss.

2.4. Experiments and evaluation

2.4.1. Comparison between different supervised losses

DINOv2 is trained with the following supervised losses: with no supervision, with CE, with HierCE, with SupCon, and with HierSupCon. Results are computed for a DINOv2 model based on ViT-L backbone.

For the self-supervised task, the model follows the DINOv2 training process with global–local loss removed, as presented by (Koch et al., 2024). In the case of hierarchical classification, the number of nodes in the last layer of the supervised head corresponds to the number of leaves in the hierarchical tree. In the case of non-hierarchical classification, only the highest degree of precision is preserved for image annotation, so coarsely labeled images are not used.

2.4.2. Evaluation of the model

Evaluation on downstream tasks. The model's feature extractor is evaluated using logistic regression (LogReg) and k-nearest neighbors (k-NN), as described in Koch et al. (Koch et al., 2024). The training images are split into 80/20 train/val ratio, which will serve to determine how many iterations are needed to obtain the best checkpoints. The validation loss reaches its minimum around 75,000 iterations, so these checkpoints are used for model testing. Features are extracted from annotated images, and LogReg and k-NN assess how labeled embeddings can classify unlabeled images.

Evaluation metrics. *Balanced Accuracy* (bAcc) are used to measure classification performance. Hierarchical metrics are also computed to verify the coherence of the latent space with the imposed hierarchy. *Hierarchical Precision* (hP) and *Hierarchical Recall* (hR) look at the overlap between the ancestors of each predicted class $A_{\text{pred},i}$ and its corresponding True class $A_{\text{True},i}$.

$$\text{hP} = \frac{\sum_{i=1}^{n}\left|\text{A}_{\text{True,i}} \cap \text{A}_{\text{pred,i}}\right|}{\sum_{i=1}^{n}\left|\text{A}_{\text{pred,i}}\right|}$$

$$\text{hR} = \frac{\sum_{i=1}^{n}\left|\text{A}_{\text{True,i}} \cap \text{A}_{\text{pred,i}}\right|}{\sum_{i=1}^{n}\left|\text{A}_{\text{True,i}}\right|}$$

The *Hierarchical F1 Score* (hF1) is computed by using hP and hR.

$$\text{hF1} = \frac{2 \bullet \text{hP} \bullet \text{hR}}{\text{hP} + \text{hR}}$$

Estimation of prediction uncertainty. For external datasets, 5-fold cross-validation is performed, which provides the mean and standard deviation for each metric.

3. Results and discussion

3.1. Classification with different supervised losses

First we analyze if cells are grouped by cell type in the latent space. k-Nearest neighbor (1NN and 20NN) and logistic regression (logreg) are performed on the embeddings of images to assess whether the models are able to correctly group images according to their classes. The results

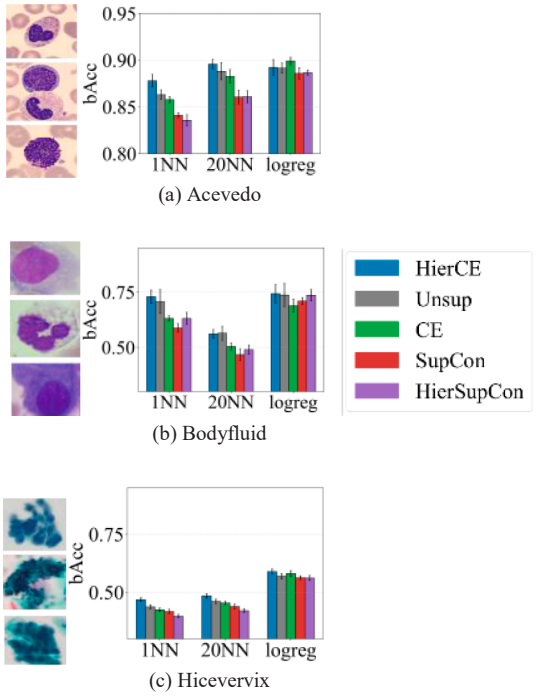


Fig. 10. bAcc for different supervised losses with Acevedo, Bodyfluid, and HiCervix datasets. The highest scores are reached by the HierCEtrained model for most downstream tasks.

are presented for the Acevedo, HiCervix, and Bodyfluid test datasets in Fig. 10.

The model trained with HierCE performs overall better by gaining percentage points on every downstream task (1NN, 20NN, LogReg) for the three external datasets. The superiority of our framework for 1NN and 20NN tasks shows that HierCE creates a better organization of cell types in the latent space for external datasets that were not encountered during training.

Interestingly, adding supervision with traditional cross entropy does not improve classification from the unsupervised model, while the addition of hierarchical cross entropy allows the model to better generalize on new datasets. This is even more striking for the HiCervix dataset, which is completely out-of domain with cervical cells: for this dataset, the addition of HierCE is beneficial for every downstream task.

3.2. Hierarchical results with different supervised losses

Secondly, we look if cell types are grouped according to the hierarchy in the latent space, i.e if clusters of cell types align with our hierarchical tree. To do so, 1NN, 20NN and LogReg are assessed with hierarchical metrics. Results are shown in Fig. 11 for Acevedo and Bodyfluid datasets.

LCA metric represents the severity of the mistake which needs to be minimized. Therefore plot 1/LCA so that the highest values of the metric represent the best results. We did not display these metrics for HiCervix, because this dataset contains cervical categories which are unrelated to our imposed hierarchy (Fig. 3).

HierCE gives the highest Hierarchical F1 and lineage accuracy both for the Acevedo and Bodyfluid datasets on each downstream task. It is coherent with the fact that the hierarchy enforces cell types to be grouped by lineage – our first hierarchy level. In Acevedo, 1/LCA reaches its maximum both for HierCE or for UnSup depending on the downstream task, and in BodyFluid for HierCE and for CE. It is therefore less evident to conclude systematic improvement with the hierarchy in the severity of mistakes.

However, it is important to mention that error bars on Fig. 11 show that the uncertainties are higher for this metric. Furthermore, most of the time HierCE is the highest 1/LCA, and if not it still comes in the top two of highest values. To verify the validity of the results, further results will be given in next sections, that well illustrate the diminution of error severity. All numerical results for can be found in Tables 1–3.

Statistical significance is also reported in Appendix B. Interestingly enough, in the rare occasions when HierCE is not the top performing model, in particular for Acevedo dataset on LogReg tasks, the difference of performance with Unsupervised model is not significant. On the opposite, the rest of the time, HierCE model is significantly better than other configuration, especially for out of domain datasets HiCervix and BodyFluid.

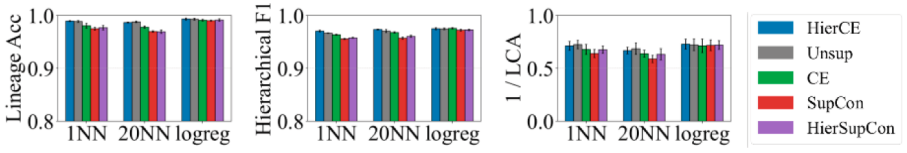
Table 1
Detailed results for Acevedo dataset for the classification metrics (wF1, bAcc) and hierarchical metrics lineage accuracy, LCA, hierarchical F1).

Model_name	Metric	wf1	bAcc
dino_L_HierCE	1NN	89.97 ± 0.61	87.80 ± 0.67
dino_L_HierCE	20NN	91.53 ± 0.24	89.58 ± 0.50
dino_L_HierCE	logreg	91.28 ± 0.55	89.23 ± 0.83
dino_L_HierSupCon	1NN	86.36 ± 0.34	83.57 ± 0.58
dino_L_HierSupCon	20NN	88.40 ± 0.41	86.08 ± 0.64
dino_L_HierSupCon	logreg	90.72 ± 0.24	88.68 ± 0.28
dino_L_CE	1NN	88.26 ± 0.16	85.78 ± 0.32
dino_L_CE	20NN	90.40 ± 0.47	88.27 ± 0.72
dino_L_CE	logreg	91.74 ± 0.22	89.90 ± 0.41
dino_L_Unsup	1NN	88.64 ± 0.50	86.31 ± 0.51
dino_L_Unsup	20NN	90.67 ± 0.62	88.82 ± 0.92
dino_L_Unsup	logreg	91.28 ± 0.35	89.15 ± 0.54
dino_L_SupCon	1NN	86.78 ± 0.20	84.10 ± 0.26
dino_L_SupCon	20NN	88.44 ± 0.48	86.05 ± 0.73
dino_L_SupCon	logreg	90.53 ± 0.43	88.59 ± 0.62
Model_name	**Lineage Acc**	**LCA**	**Hierarchical F1**
dino_L_HierCE	98.94 ± 0.09	1.40 ± 0.04	97.04 ± 0.16
dino_L_HierCE	98.67 ± 0.09	1.50 ± 0.03	97.35 ± 0.08
dino_L_HierCE	99.33 ± 0.20	1.37 ± 0.05	97.49 ± 0.20
dino_L_HierSupCon	97.69 ± 0.43	1.48 ± 0.03	95.77 ± 0.09
dino_L_HierSupCon	96.91 ± 0.33	1.58 ± 0.05	96.05 ± 0.22
dino_L_HierSupCon	99.14 ± 0.25	1.39 ± 0.04	97.26 ± 0.15
dino_L_CE	98.06 ± 0.48	1.47 ± 0.05	96.35 ± 0.10
dino_L_CE	97.79 ± 0.22	1.56 ± 0.03	96.78 ± 0.16
dino_L_CE	99.10 ± 0.18	1.40 ± 0.06	97.56 ± 0.14
dino_L_Unsup	98.86 ± 0.20	1.38 ± 0.04	96.65 ± 0.07
dino_L_Unsup	98.79 ± 0.14	1.46 ± 0.05	97.05 ± 0.29
dino_L_Unsup	99.30 ± 0.18	1.38 ± 0.05	97.46 ± 0.16
dino_L_SupCon	97.49 ± 0.28	1.56 ± 0.04	95.55 ± 0.11
dino_L_SupCon	96.96 ± 0.16	1.70 ± 0.03	95.70 ± 0.21
dino_L_SupCon	99.02 ± 0.11	1.39 ± 0.05	97.21 ± 0.18

3.3. UMAP embeddings

The representation of the embeddings from the Acevedo dataset is shown using UMAP projection in Fig. 12.

The quantitative improvements observed in the bAcc metrics are apparent in the UMAP plot: the model trained with HierCE appears to produce more compact clusters in the Acevedo dataset and better separates distant classes in the hierarchy (Fig. 12). In particular, the granulocyte lineage is more clearly divided into immature (promyelocyte, myelocyte, metamyelocyte) and mature (band neutrophil and segmented neutrophil) stages. Moreover, monocytes belonging to a distinct branch of the hierarchy are positioned further apart from the other cell types.



(a) Acevedo

(b) Bodyfluid

Fig. 11. Lineage accuracy, LCA, and Hierarchical F1 metrics for different supervised losses with Acevedo and Bodyfluid datasets. The HierCE trained model gives the highest Hierarchical F1 and Lineage accuracy. The highest 1/LCA is obtained with the HierCE, CE, or with the unsupervised mode.

Table 2
Detailed results for BodyFluid dataset for the classification metrics (wF1, bAcc) and hierarchical metrics lineage accuracy, LCA, hierarchical F1).

Model_name	Metric	wf1	bAcc
dino_L_HierCE	1NN	88.32 ± 0.59	72.96 ± 3.01
dino_L_HierCE	20NN	89.24 ± 0.79	56.08 ± 1.96
dino_L_HierCE	logreg	90.73 ± 0.81	74.31 ± 4.16
dino_L_HierSupCon	1NN	86.46 ± 0.73	63.11 ± 2.75
dino_L_HierSupCon	20NN	86.00 ± 0.63	48.97 ± 2.13
dino_L_HierSupCon	logreg	90.31 ± 0.40	73.61 ± 2.57
dino_L_CE	1NN	86.32 ± 0.47	63.16 ± 1.23
dino_L_CE	20NN	87.44 ± 0.39	50.38 ± 1.64
dino_L_CE	logreg	90.04 ± 0.57	68.92 ± 2.88
dino_L_Unsup	1NN	88.64 ± 0.71	70.75 ± 5.38
dino_L_Unsup	20NN	89.29 ± 0.97	56.42 ± 3.16
dino_L_Unsup	logreg	90.69 ± 0.33	73.71 ± 5.21
dino_L_SupCon	1NN	85.48 ± 0.49	58.78 ± 2.03
dino_L_SupCon	20NN	85.06 ± 0.67	46.72 ± 2.69
dino_L_SupCon	logreg	89.83 ± 0.56	70.99 ± 1.54
Model_name	**Lineage Acc**	**LCA**	**Hierarchical F1**
dino_L_HierCE	79.16 ± 2.57	2.77 ± 0.05	93.17 ± 0.32
dino_L_HierCE	73.08 ± 2.24	2.82 ± 0.06	93.87 ± 0.51
dino_L_HierCE	80.43 ± 1.69	2.78 ± 0.05	94.51 ± 0.48
dino_L_HierSupCon	70.97 ± 2.18	2.84 ± 0.06	91.83 ± 0.53
dino_L_HierSupCon	67.45 ± 2.55	2.84 ± 0.06	92.21 ± 0.36
dino_L_HierSupCon	80.14 ± 1.11	2.80 ± 0.04	94.25 ± 0.31
dino_L_CE	72.86 ± 1.86	2.83 ± 0.05	91.77 ± 0.39
dino_L_CE	66.78 ± 2.69	2.88 ± 0.05	92.81 ± 0.24
dino_L_CE	77.43 ± 1.97	2.71 ± 0.06	94.13 ± 0.38
dino_L_Unsup	76.81 ± 1.78	2.79 ± 0.08	93.10 ± 0.39
dino_L_Unsup	71.31 ± 3.05	2.84 ± 0.09	93.92 ± 0.47
dino_L_Unsup	79.48 ± 1.92	2.79 ± 0.10	94.46 ± 0.22
dino_L_SupCon	67.44 ± 1.76	2.82 ± 0.06	91.41 ± 0.33
dino_L_SupCon	59.80 ± 1.73	2.85 ± 0.07	91.65 ± 0.41
dino_L_SupCon	77.25 ± 1.98	2.75 ± 0.07	94.05 ± 0.43

Table 3
Detailed results for HiCervix dataset for the classification metrics (wF1, bAcc).

Model name	Metric	wf1	bAcc
dino_L_HierCE	1NN	47.17 ± 0.96	46.86 ± 0.77
dino_L_HierCE	20NN	49.47 ± 1.06	48.47 ± 0.97
dino_L_HierCE	logreg	58.85 ± 1.19	58.93 ± 1.12
dino_L_HierSupCon	1NN	41.20 ± 0.79	39.74 ± 0.78
dino_L_HierSupCon	20NN	43.20 ± 0.93	42.14 ± 0.79
dino_L_HierSupCon	logreg	56.86 ± 0.81	56.19 ± 1.08
dino_L_CE	1NN	43.54 ± 0.79	42.51 ± 0.79
dino_L_CE	20NN	46.10 ± 0.82	45.44 ± 0.77
dino_L_CE	logreg	58.60 ± 0.93	58.12 ± 1.23
dino_L_Unsup	1NN	44.42 ± 1.17	43.80 ± 0.89
dino_L_Unsup	20NN	47.05 ± 1.09	46.12 ± 0.95
dino_L_Unsup	logreg	57.82 ± 1.17	56.95 ± 1.20
dino_L_SupCon	1NN	42.30 ± 0.66	41.81 ± 0.97
dino_L_SupCon	20NN	43.93 ± 1.29	43.95 ± 1.05
dino_L_SupCon	logreg	57.43 ± 0.69	56.32 ± 0.80

3.4. Confusion matrices

To better understand the behaviour of the model in case of mistakes, confusion matrices are plotted for Acevedo dataset predictions in Fig. 13.

The eosinophil class provides a good illustration of the model making less severe mistakes when trained hierarchically. The HierCE-trained model no longer predicts eosinophils as erythroblasts, since these classes belong to different lineages. Instead, most remaining errors involve predicting eosinophils as segmented neutrophils, which share the same lineage.

Similarly, when using the hierarchy, metamyelocytes are more often confused with promyelocytes, a very closely related class, rather than being predicted as band neutrophils.

Therefore, although 1/LCA shows variability, the confusion matrices confirm that HierCE reduces cross-lineage errors, making algorithm mistakes more biologically acceptable. Finally, the previous hierarchical results, we mentioned an improvement in lineage accuracy across models, which further demonstrates the reduction in mistake severity, since lineage is particularly important for biological applications.

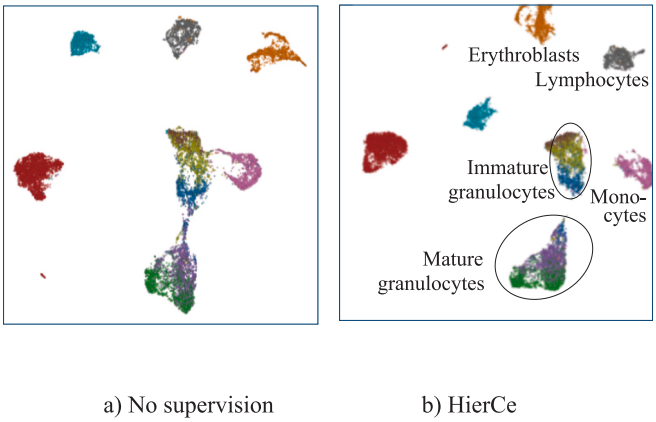


Fig. 12. HierCE creates more compact clusters, aligned with hierarchy groupings. UMAPs of Acevedo embeddings for model trained without supervision (left) and with HierCE (right).

3.5. Class imbalance in hierarchical training

In hierarchical training, class imbalance can occur at multiple levels: *across branches*, *within branches* (i.e., between sibling classes), and *across depths* when fine-grained labels are rare. When using the HierCE loss, each sample is represented by all its ancestors, which helps mitigate imbalance *across depths*. In addition, giving supervision to ancestor nodes allows rare classes to receive learning signals from their siblings, reducing the impact of *within-branch imbalance*.

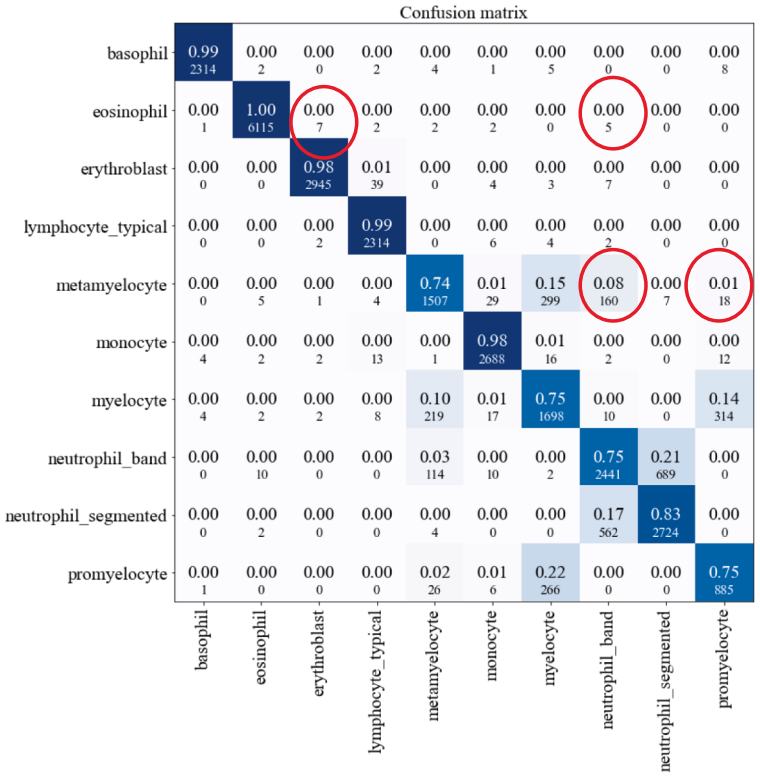
However, in HierCE training, majority branches can still dominate the learning process. One possible strategy to address this issue is to assign different weights to branches. Currently, weights depend only on the depth of a node in the hierarchy, but they could also be adjusted based on the overall proportion of a branch within the tree, for example with the number of samples belonging to the leaf classes under that branch. In our case, such a strategy was not necessary, as the class distribution across the hierarchy was overall relatively balanced.

3.6. Ablation study on the parameter α

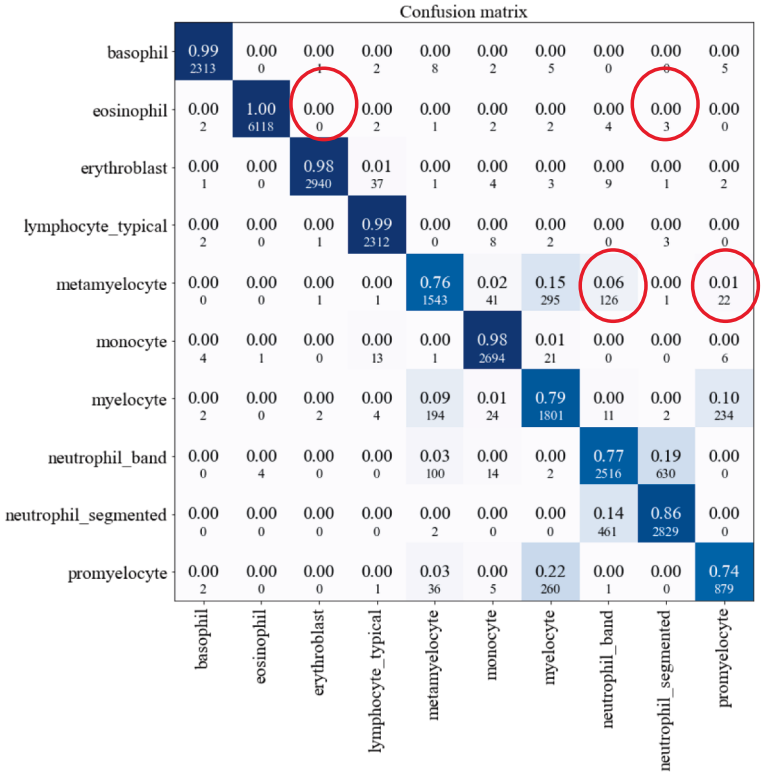
In this part, the influence of the hyperparameter α in HierCE computation is studied. This parameter changes how information flows through the hierarchy, affecting whether the model prioritizes high-level or low-level classification. In particular, it is responsible for the severity of mistakes, by imposing predictions to be closer to the actual class in the hierarchy. Three models are trained with HierCE using different α parameters (0.1, 0.5, 0.9), and their embedding spaces are compared using classification and hierarchical metrics. The comparison is done on embeddings from Acevedo dataset. This is because this dataset contains images from blood samples, which are visually closer to those from the training. It means their representation will more closely align the hierarchy, allowing the influence of the parameter α to be better isolated. The results are reported in Fig. 14.

Using the balanced accuracy metric (Fig. 14a), results are very similar for the different values of α. This could be expected, since this metric reveals flat classification performance and does not take into consideration hierarchical misclassification. If bAcc is not influenced by modifications of α, it means that the hierarchy is well aligned with the data.

In the case of hierarchical metrics in Fig. 14b (Lineage Accuracy,

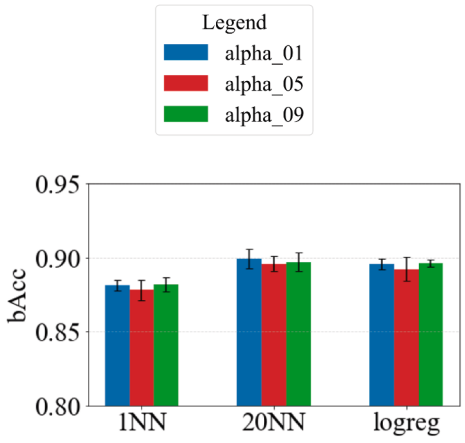


a) Unsupervised

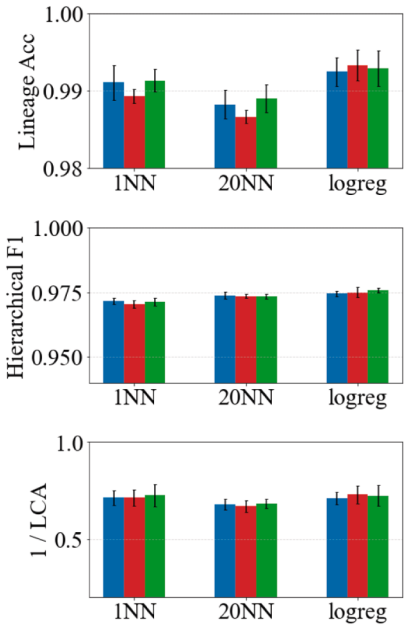


b) HierCE

Fig. 13. Confusion matrices of unsupervised and HierCE trained models. HierCE model mistakes are less severe than unsupervised model. Notable differences between model mistakes are circled in red.



a) Comparison of classification tasks with balanced accuracy for different α parameters.



b) Comparison of classification tasks with hierarchical metrics for different α parameters.

Fig. 14. Evaluation of the influence of the α parameter on the embedding space, by evaluating kNN and LogReg performances with clasification and hierarchical metrics.

Hierarchical F1, 1/LCA), the behaviour of the metric depends on wether classification is done with kNN or LogReg. It is interesting first to mention that kNN is a local metric probe: it is therefore influenced by clear separation at different hierarchy levels. On the opposite, logistic regression is a global probe. It might be more sensitive to overall better alignment to the hierarchy, regardless of the level.

For hierarchical metrics, $\alpha = 0.5$ yields higher LogReg results. However, kNN classification task is better for the values $\alpha = 0.1$ and $\alpha = 0.9$. A likely explanation for kNN higher accuracy could be that local neighborhood structure is better preserved when hierarchical supervision is concentrated at a single scale (high or low level) rather than distributed across level. On the opposite, $\alpha = 0.5$ encourages representations in which information is more evenly distributed across hierarchical levels, and therefore advantages LogReg global alignment.

We choose $\alpha = 0.5$ because linear probe better represents semantic representation of the classes, and is therefore more likely to generalize better to unknown downstream tasks. This is supported by the fact that $\alpha = 0.5$ yields better results in (Bertinetto et al., 2020) for numerous large datasets, including tieredImageNet-H which contains 608 classes distributed in a 13-level hierarchy.

3.7. Comparison with other hierarchical methods

It is worth noting that other methods exist that use a hierarchical structure for training the model. Therefore, to provide a fair justification for the choice of HierCE, we compare the results with two widely used frameworks: multilevel classification and distance-aware classification.

3.7.1. Multilevel classification

It is possible to train the model using different classification losses corresponding to each level of the hierarchy, from coarse to fine. Each loss can be weighted by a factor that penalizes the levels differently (Muller & Smith, 2020). In our case, we use a multi-head MLP classifier, where each head corresponds to the classification of one level of the hierarchy. Cross-entropy loss is applied to each head, and the final loss is obtained by averaging them.

3.7.2. Distance aware classification

Another solution is to include a distance-aware loss. The hierarchy provides distances between each class, and the loss penalizes mistakes according to their distance to the real class. This loss acts as a form of label smoothing where where misclassifications are penalized proportionally to their tree distance. Semantic similarity is an appropriate loss function for distance-aware optimization. The final loss is defined as follows:

$$L = \sum_{c=1}^{C} p_c \bullet D[y, c]$$

With p the predicted probabilities, and y the True class. For example $D[c, c] = 0$.

3.7.3. Comparison results

The results are shown in Fig. 15.

Again, results are compared on Acevedo dataset, which more closely aligns the hierarchy. Overall, the three hierarchical frameworks are comparable in terms of performance. However, it seems that generally HierCE outperforms the other two frameworks.

This could be explained by an important factor: HierCE produces coherent gradients across hierachy levels. By taking into consideration the whole path from the root to the predicted node, it will learn that probabilities are conditioned by their ancestor. Therefore classes are not considered independantly, but instead are predicted from a distribution of nested probabilities over the entire tree.

On the opposite, when using multilevel cross entropy, each head sees a different classification problem. Thus, it does not enforce consistency between levels. Even if this method provides a good discrimation per level, it does not structure embeddings according to the entire hierarchy as explicitely as HierCE does. Distance-based loss is also a weak representation of the hierarchy. It structures the latent space with distances, but does not explicitly push embeddings away from incorrect branches. This loss is actually more suited to ranking classes in the correct order than to clustering them according to different granularities. It is also important to mention that distances here are defined uniquely by the tree structure and do not represent well the biology of the cells. For example, the biological difference between two siblings in the tree may vary with their cell type, and distance should be defined accordingly. However, computing biological distances between classes is a non-trivial task and would require a multivariate analysis.

Additionally to these elements of comparison, HierCE also presents inherent advantages. First, the fact that it learns a probabilistic representation of each category makes it more adaptable, for example, when

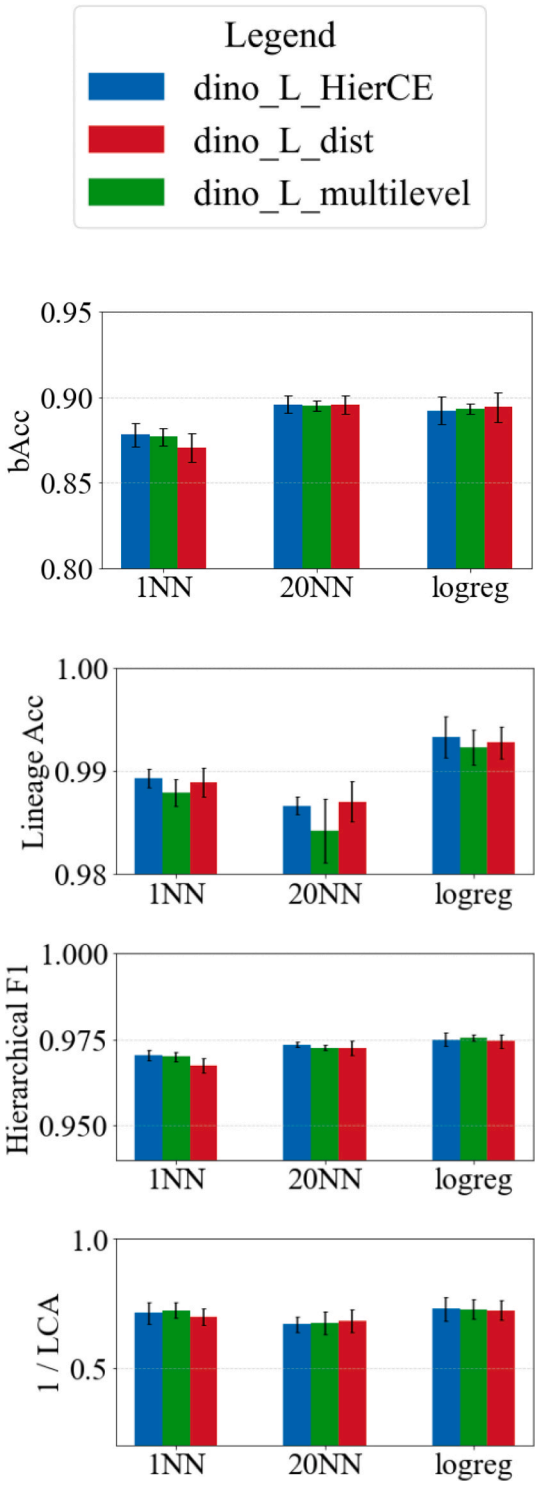


Fig. 15. Comparison between different hierarchical strategies.HierCE overall outperforms the two other frameworks.

introducing a new category. Second, it allows flexible tuning of the communication between each level through the adjustment of the weights λ, providing control over how mistakes are penalized at different depths in the hierarchy and preserving partial credit for correct higher-level predictions.

Finally, the most important reason for our choice is to keep the structure of the loss aligned as much as possible with traditional loss term Cross Entropy. Hence, we better isolate and understand the influence of the hierarchy, as training conditions remain nearly identical except for the hierarchical formulation.

3.8. Discussion

DINOv2-based approaches are particularly powerful for clustering cells by type without requiring individual labeling of each cell during training. In this paper, it is shown that adding hierarchical supervision to the self-supervised DINOv2 model further enhances the clustering of cell types for external datasets, even for datasets containing cells from different tissues such as body fluid or cervical images.

Introducing hierarchy through HierCE has advantages: it integrates expert knowledge into the model and thereby produces a biologically relevant representation of cell types. Cells are better grouped by lineage which has a clinical meaning. Additionally, it allows training with labels of different granularities. As a result, hierarchy improves performance on external datasets and reduces the severity of misclassifications.

Despite its contributions, this study is not without limitations. Self-supervised methods are developed in a less constrained setting than supervised ones. More precisely, the model is trained to perform on an auxiliary task: its aim is to obtain a meaningful representation of the cells, from which downstream tasks can be applied. Here, DINOv2 has three objectives: classifying the cells, reconstructing masked parts of the image, and creating similar embeddings for augmented versions of the same image. Therefore, the loss is not straightforward: an optimized DINOv2 will not necessary be top-performer at classification. On the other hand, interpretability mostly relies on finding the pixels, features or images that have a strong influence on model high or low performance. For a model trained on an auxiliary task such as DINOv2, we could understand the impact of a feature on the task performance, but it will not necessarily inform us about the importance of the feature for our task of interest, in this case classification from the embedding space. Therefore, understanding the underlying mechanisms of the model is particularly challenging.

In this work, hierarchical training is studied in combination with self-supervised learning, so the effects of hierarchy are intrinsically connected to self-supervised training. In addition, starting from self-supervised DINO-Bloom weights further accentuates this phenomenon. The aim of this approach is more to understand the contribution of adding hierarchical training in the traditional DINOv2 pipeline than to understand the internal mechanisms of hierarchy itself. Therefore, it would be interesting to study the relative importance between the classification task and self-supervised tasks by independently training the model for hierarchical classification without a self-supervised objective. Analyzing the influence of hierarchy in a traditional classifier model could be a starting point.

It is also important to underline that DINOv2 is a high-resource model. Effective training requires at least hundreds of thousands of images, and the model demands substantial GPU resources to achieve a reasonable training time. In our case, in particular, two A100 80 GB GPUs were needed to train DINOv2 Large, which contains 300 million parameters. Such large architectures, together with correspondingly large training datasets, are very useful for achieving sufficient generalisability. Alternatively, DINOv2 is available in varying sizes—namely small, base, large, and giant—with performance generally increasing with model size (Koch et al., 2024). Future work could explore the trade-off between model size and performance, as well as study the influence of the amount of training data on generalisability.

4. Conclusion

Hierarchical supervision enhances how images are represented in the model's latent space and leads to improved performance on various downstream tasks for three external datasets. Importantly, one of the datasets contains cervical cells, which have highly different aspects, further enhancing the out-of-domain capacities of this framework.

Additionally, biological knowledge is introduced by representing cell

classes in a hierarchy. This enhances the biological relevance of the latent space, which is measured with hierarchical F1 score, lineage accuracy, and LCA measurements for the assessment of mistake severity.

This study is the first to include prior cytology knowledge in the training of a DINOv2-based model. By increasing the model's robustness and clinical relevance, it aligns with the clinical objectives of AI applied to medicine. Nonetheless, further work is needed to better understand the mechanisms of DINOv2-based models and hierarchical semi-supervised learning in general.

CRediT authorship contribution statement

Manon Chossegros: Conceptualization, Formal analysis, Investigation, Methodology, Visualization, Writing – original draft. **Sophia Wagner:** Conceptualization, Investigation, Methodology, Validation, Visualization, Writing – review & editing. **Christian Matek:** Conceptualization, Data curation, Resources, Validation, Writing – review & editing. **Daniel Stockholm:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Writing – review & editing. **Xavier Tannier:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Writing – review & editing. **Carsten Marr:** Conceptualization, Formal analysis, Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Detailed labels

..

Table A.1
Description of the classes for datasets when they are available. HiCervix classes are given in (Cai et al., 2024).

Dataset	Supervision Classes
AML Matek	basophil, eosinophil, erythroblast, lymphocyte typical, lymphocyte atypical, metamyelocyte, monoblast, monocyte, myeloblast, myelocyte, neutrophil band, neutrophil segmented, promyelocyte, promyelocyte bilobed, smudge cell
LISC	basophil, eosinophil, lymphocyte mature, monocyte, neutrophil
Raabin	eosinophil, lymphocyte mature, monocyte, neutrophil
BMC	basophil, blast, eosinophil, eosinophil abnormal, erythroblast, fagott cell, hairy cell, lymphocyte immature, lymphocyte typical, metamyelocyte, monocyte, myelocyte, neutrophil band, neutrophil segmented, plasma cell, proerythroblast, promyelocyte, smudge cell, artefact, other, not identified
MLL23	promyelocyte atypical, basophil, eosinophil, erythroblast, hairy cell, lymphocyte large granular, lymphocyte neoplastic, lymphocyte reactive, lymphocyte typical, metamyelocyte, monoblast, myeloblast, myelocyte, neutrophil band, neutrophil segmented, plasma cell, promyelocyte, smudge cell
AML Hehr	patient classes
Warty pig	basophil, eosinophil, monocyte, neutrophil
Acevedo	basophil, eosinophil, erythroblast, lymphocyte typical, metamyelocyte, monocyte, myelocyte, neutrophil band, neutrophil segmented, promyelocyte
BodyFluid	mesothelial cell, abnormal mononuclear cell, neutrophil, basophil, mitotic cells, promonocyte, atypical lymphocyte, plasma cell, monocyte, lymphocyte, cells stained dark blue with fused boundaries,eosinophil, signet ring cell, macrophage
Hicervix	HSV, Normal, MPC, LSIL, EMC, FUNGI, ASC-H, SCC, HCG, HSIL, AGC, ASC-US, AGC-NOS, Atrophy, ACTINO, PG, ADC, ECC, AGC-ECC-NOS, CC, RPC, TRI, AGC-FN, ADC-ECC, AGC-EMCNOS, ADC-EMC

Appendix B. Statistical significance of model comparisons

We tested whether the HierCE trained model differs significantly from other models using a stratified bootstrap approach. For each bootstrap sample, we resampled the data within each class (so class proportions stayed the same), computed the chosen performance metric for both models, and recorded the difference between them (model B − HierCE model).

The observed effect is the metric difference calculated on the original dataset. We then built a two-sided 95% confidence interval for this difference using the 2.5th and 97.5th percentiles of the bootstrap differences. If this confidence interval does not include zero, we consider the difference statistically significant.

1. Acevedo

Metric	Other	HierCE	Significant	Metric worse	metric better	delta	CI low	CI high	Task
wf1	HierSupCon	HierCE	True	0.8637	0.8998	0.0361	0.0319	0.0386	1nn
wf1	SupCon	HierCE	True	0.8679	0.8998	0.0319	0.0268	0.0349	1nn
wf1	CE	HierCE	True	0.8827	0.8998	0.0171	0.0130	0.0209	1nn
wf1	Unsup	HierCE	True	0.8864	0.8998	0.0133	0.0104	0.0179	1nn
wf1	HierSupCon	HierCE	True	0.8841	0.9153	0.0312	0.0292	0.0327	20nn
wf1	SupCon	HierCE	True	0.8845	0.9153	0.0308	0.0260	0.0333	20nn
wf1	CE	HierCE	True	0.9041	0.9153	0.0112	0.0055	0.0141	20nn
wf1	Unsup	HierCE	True	0.9067	0.9153	0.0086	0.0058	0.0103	20nn
wf1	SupCon	HierCE	True	0.9054	0.9129	0.0075	0.0033	0.0118	logreg
wf1	HierSupCon	HierCE	True	0.9072	0.9129	0.0056	0.0003	0.0085	logreg
wf1	Unsup	HierCE	False	0.9128	0.9129	0.0000	−0.0041	0.0026	logreg
bacc	HierSupCon	HierCE	True	0.8357	0.8780	0.0423	0.0362	0.0465	1nn
bacc	SupCon	HierCE	True	0.8410	0.8780	0.0370	0.0308	0.0410	1nn

(continued on next page)

(*continued*)

Metric	Other	HierCE	Significant	Metric worse	metric better	delta	CI low	CI high	Task
bacc	CE	HierCE	True	0.8578	0.8780	0.0202	0.0151	0.0246	1nn
bacc	Unsup	HierCE	True	0.8631	0.8780	0.0149	0.0111	0.0225	1nn
bacc	SupCon	HierCE	True	0.8605	0.8958	0.0353	0.0300	0.0376	20nn
bacc	HierSupCon	HierCE	True	0.8608	0.8958	0.0349	0.0328	0.0386	20nn
bacc	CE	HierCE	True	0.8827	0.8958	0.0131	0.0061	0.0168	20nn
bacc	Unsup	HierCE	True	0.8882	0.8958	0.0076	0.0033	0.0100	20nn
bacc	SupCon	HierCE	True	0.8859	0.8923	0.0064	0.0009	0.0103	logreg
bacc	HierSupCon	HierCE	False	0.8868	0.8923	0.0055	−0.0020	0.0090	logreg
bacc	Unsup	HierCE	False	0.8915	0.8923	0.0008	−0.0043	0.0032	logreg
lca	Unsup	HierCE	False	1.3769	1.4021	0.0252	−0.0098	0.0611	1nn
lca	CE	HierCE	True	1.4722	1.4021	−0.0701	−0.0887	−0.0368	1nn
lca	HierSupCon	HierCE	True	1.4766	1.4021	−0.0745	−0.0966	−0.0536	1nn
lca	Unsup	HierCE	True	1.4594	1.4954	0.0360	0.0075	0.0635	20nn
lca	CE	HierCE	True	1.5643	1.4954	−0.0689	−0.1119	−0.0259	20nn
lca	HierSupCon	HierCE	True	1.5801	1.4954	−0.0847	−0.1229	−0.0571	20nn
lca	Unsup	HierCE	False	1.3860	1.3743	−0.0117	−0.0258	0.0152	logreg
lca	HierSupCon	HierCE	False	1.3873	1.3743	−0.0130	−0.0609	0.0421	logreg
lca	SupCon	HierCE	False	1.3939	1.3743	−0.0196	−0.0645	0.0396	logreg
hf1	SupCon	HierCE	True	0.9555	0.9704	0.0149	0.0129	0.0163	1nn
hf1	HierSupCon	HierCE	True	0.9577	0.9704	0.0127	0.0110	0.0141	1nn
hf1	CE	HierCE	True	0.9635	0.9704	0.0069	0.0051	0.0084	1nn
hf1	Unsup	HierCE	True	0.9665	0.9704	0.0039	0.0029	0.0058	1nn
hf1	SupCon	HierCE	True	0.9570	0.9735	0.0165	0.0158	0.0171	20nn
hf1	HierSupCon	HierCE	True	0.9605	0.9735	0.0130	0.0118	0.0138	20nn
hf1	CE	HierCE	True	0.9678	0.9735	0.0057	0.0037	0.0070	20nn
hf1	Unsup	HierCE	True	0.9705	0.9735	0.0030	0.0013	0.0040	20nn
hf1	SupCon	HierCE	True	0.9721	0.9749	0.0028	0.0014	0.0032	logreg
hf1	HierSupCon	HierCE	True	0.9726	0.9749	0.0024	0.0006	0.0035	logreg
hf1	Unsup	HierCE	False	0.9746	0.9749	0.0003	−0.0011	0.0009	logreg
lin_acc	SupCon	HierCE	True	0.9749	0.9894	0.0144	0.0123	0.0170	1nn
lin_acc	HierSupCon	HierCE	True	0.9769	0.9894	0.0124	0.0087	0.0148	1nn
lin_acc	CE	HierCE	True	0.9806	0.9894	0.0088	0.0057	0.0102	1nn
lin_acc	Unsup	HierCE	False	0.9886	0.9894	0.0008	−0.0013	0.0033	1nn
lin_acc	HierSupCon	HierCE	True	0.9691	0.9867	0.0175	0.0139	0.0185	20nn
lin_acc	SupCon	HierCE	True	0.9696	0.9867	0.0170	0.0152	0.0206	20nn
lin_acc	CE	HierCE	True	0.9779	0.9867	0.0088	0.0058	0.0117	20nn
lin_acc	SupCon	HierCE	True	0.9902	0.9933	0.0032	0.0010	0.0057	logreg
lin_acc	CE	HierCE	True	0.9910	0.9933	0.0023	0.0008	0.0033	logreg
lin_acc	HierSupCon	HierCE	True	0.9914	0.9933	0.0019	0.0006	0.0043	logreg
lin_acc	Unsup	HierCE	False	0.9930	0.9933	0.0004	−0.0013	0.0008	logreg
lin_acc	SupCon	Unsup	True	0.9696	0.9879	0.0182	0.0146	0.0207	20nn
lin_acc	CE	Unsup	True	0.9779	0.9879	0.0100	0.0063	0.0119	20nn
lin_acc	HierCE	Unsup	False	0.9867	0.9879	0.0012	−0.0007	0.0025	20nn
lin_acc	SupCon	HierCE	True	0.9902	0.9933	0.0032	0.0017	0.0057	logreg
lin_acc	CE	HierCE	True	0.9910	0.9933	0.0023	0.0007	0.0047	logreg
lin_acc	HierSupCon	HierCE	True	0.9914	0.9933	0.0019	0.0004	0.0049	logreg
lin_acc	Unsup	HierCE	False	0.9930	0.9933	0.0004	−0.0008	0.0023	logreg

2. BodyFluid

Metric	Other	HierCE	Significant	Metric worse	Metric better	Delta	CI low	CI high	Task
wf1	SupCon	HierCE	True	0.8548	0.8832	−0.5839	−0.5866	−0.5827	1nn
wf1	CE	HierCE	True	0.8632	0.8832	−0.5919	−0.5944	−0.5886	1nn
wf1	HierSupCon	HierCE	True	0.8645	0.8832	−0.5886	−0.5899	−0.5868	1nn
wf1	SupCon	HierCE	True	0.8507	0.8925	−0.5793	−0.5834	−0.5742	20nn
wf1	HierSupCon	HierCE	True	0.8601	0.8925	−0.5829	−0.5853	−0.5804	20nn
wf1	CE	HierCE	True	0.8745	0.8925	−0.6007	−0.6041	−0.5967	20nn
wf1	SupCon	HierCE	True	0.8533	0.9074	−0.5790	−0.5828	−0.5747	logreg
wf1	Unsup	HierCE	True	0.8903	0.9074	−0.6168	−0.6199	−0.6118	logreg
wf1	CE	HierCE	True	0.9004	0.9074	−0.6321	−0.6361	−0.6305	logreg
wf1	HierSupCon	HierCE	True	0.9031	0.9074	−0.6310	−0.6342	−0.6285	logreg
bacc	SupCon	HierCE	True	0.5904	0.7260	−0.5195	−5.26E-01	−0.5163	1nn
bacc	CE	HierCE	True	0.6305	0.7260	−0.5581	−0.5634	−0.5512	1nn
bacc	HierSupCon	HierCE	True	0.6311	0.7260	−0.5582	−0.5633	−0.5526	1nn
bacc	Unsup	HierCE	True	0.7089	0.7260	−0.6360	−0.6421	−0.6275	1nn
bacc	SupCon	HierCE	True	0.4632	0.5604	−0.3937	−0.3963	−0.3886	20nn
bacc	HierSupCon	HierCE	True	0.4894	0.5604	−0.4174	−0.4253	−0.4141	20nn
bacc	CE	HierCE	True	0.5028	0.5604	−0.4311	−0.4380	−0.4228	20nn
bacc	SupCon	HierCE	True	0.5268	0.7437	−0.4391	−0.4516	−0.4321	logreg
bacc	Unsup	HierCE	True	0.6366	0.7437	−0.5489	−0.5543	−0.5385	logreg
bacc	CE	HierCE	True	0.6892	0.7437	−0.6193	−0.6239	−0.6100	logreg

(*continued on next page*)

(*continued*)

Metric	Other	HierCE	Significant	Metric worse	Metric better	Delta	CI low	CI high	Task
bacc	HierSupCon	HierCE	True	0.7332	0.7437	−0.6626	−0.6735	−0.6539	logreg
lca	HierSupCon	HierCE	True	3.0000	3.1180	0.6895	0.6666	0.7016	1nn
lca	SupCon	HierCE	True	3.0500	3.1180	0.6460	0.6044	0.6760	1nn
lca	Unsup	HierCE	True	3.0762	3.1180	0.6244	0.5835	0.6636	1nn
lca	CE	HierCE	True	3.0410	3.0418	0.6712	0.6446	0.6926	20nn
lca	Unsup	HierCE	True	3.0520	3.0418	0.6575	0.6271	0.6748	20nn
lca	SupCon	HierCE	True	3.1736	3.0418	0.5374	0.5072	0.5647	20nn
lca	HierSupCon	HierCE	True	2.9914	3.1479	0.7094	0.6942	0.7341	logreg
lca	CE	HierCE	True	3.0059	3.1479	0.6920	0.6707	0.7199	logreg
lca	Unsup	HierCE	True	3.0650	3.1479	0.6617	0.6377	0.6869	logreg
lca	SupCon	HierCE	True	3.1132	3.1479	0.5767	0.5600	0.6187	logreg
hf1	SupCon	HierCE	True	0.9615	0.9632	−0.5031	−0.5072	−0.5000	1nn
hf1	CE	HierCE	True	0.9634	0.9632	−0.5018	−0.5067	−0.4968	1nn
hf1	HierSupCon	HierCE	True	0.9657	0.9632	−0.4981	−0.4993	−0.4962	1nn
hf1	SupCon	HierCE	True	0.9575	0.9698	−0.4944	−0.4995	−0.4898	20nn
hf1	HierSupCon	HierCE	True	0.9633	0.9698	−0.4970	−0.5006	−0.4946	20nn
hf1	CE	HierCE	True	0.9699	0.9698	−0.5056	−0.5086	−0.5013	20nn
hf1	SupCon	HierCE	True	0.9595	0.9703	−0.4957	−0.5004	−0.4923	logreg
hf1	CE	HierCE	True	0.9688	0.9703	−0.5119	−0.5174	−0.5078	logreg
hf1	HierSupCon	HierCE	True	0.9705	0.9703	−0.5099	−0.5147	−0.5065	logreg
lin_acc	SupCon	HierCE	True	0.8943	0.9021	−0.5621	−0.5673	−0.5599	1nn
lin_acc	HierSupCon	HierCE	True	0.8979	0.9021	−0.5631	−0.5662	−0.5603	1nn
lin_acc	CE	HierCE	True	0.9042	0.9021	−0.5717	−0.5796	−0.5647	1nn
lin_acc	SupCon	HierCE	True	0.8957	0.9212	−0.5644	−0.5744	−0.5567	20nn
lin_acc	HierSupCon	HierCE	True	0.9005	0.9212	−0.5656	−0.5699	−0.5612	20nn
lin_acc	CE	HierCE	True	0.9268	0.9212	−0.5937	−0.5995	−0.5862	20nn
lin_acc	SupCon	HierCE	True	0.8951	0.9303	−0.5558	−0.5612	−0.5526	logreg
lin_acc	HierSupCon	HierCE	True	0.9225	0.9303	−0.5899	−0.5975	−0.5860	logreg
lin_acc	CE	HierCE	True	0.9253	0.9303	−0.5969	−0.6026	−0.5926	logreg
lin_acc	Unsup	HierCE	True	0.9274	0.9303	−0.5943	−0.6000	−0.5871	logreg
	dino_L_CE	dino_L_Unsup	True	0.9768	0.9808	0.0080	0.0069	0.0107	1nn
	dino_L_HierCE	dino_L_Unsup	True	0.9778	0.9808	−0.0008	−0.0018	−0.0002	1nn
lineage_acc	dino_L_SupCon	dino_L_Unsup	True	0.9772	0.9857	0.0182	0.0146	0.0207	20nn
lineage_acc	dino_L_HierCE	dino_L_Unsup	False	0.9818	0.9857	0.0012	−0.0007	0.0025	20nn
lineage_acc	dino_L_HierSupCon	dino_L_Unsup	True	0.9830	0.9857	0.0187	0.0124	0.0207	20nn
lineage_acc	dino_L_CE	dino_L_Unsup	True	0.9841	0.9857	0.0100	0.0063	0.0119	20nn
lineage_acc	dino_L_Unsup	dino_L_CE	True	0.9781	0.9853	−0.0019	−0.0037	−0.0002	logreg
lineage_acc	dino_L_HierCE	dino_L_CE	True	0.9800	0.9853	−0.0023	−0.0047	−0.0007	logreg
lineage_acc	dino_L_SupCon	dino_L_CE	False	0.9802	0.9853	0.0009	−0.0013	0.0045	logreg
lineage_acc	dino_L_HierSupCon	dino_L_CE	False	0.9819	0.9853	−0.0004	−0.0020	0.0026	logreg

3. HiCervix

Metric	Other	HierCE	Significant	Metric worse	Metric better	Delta	CI low	CI high	Task
wf1	HierSupCon	HierCE	True	0.4120	0.4718	−0.3652	−0.3672	−0.3633	1nn
wf1	CE	HierCE	True	0.4353	0.4718	−0.3889	−0.3920	−0.3861	1nn
wf1	HierSupCon	HierCE	True	0.4320	0.4947	−0.3875	−0.3892	−0.3841	20nn
wf1	CE	HierCE	True	0.4610	0.4947	−0.4142	−0.4147	−0.4121	20nn
wf1	HierSupCon	HierCE	True	0.4716	0.5189	−0.4250	−0.4266	−0.4232	logreg
wf1	SupCon	HierCE	True	0.5743	0.5189	−0.5277	−0.5289	−0.5240	logreg
bacc	HierSupCon	HierCE	True	0.3976	0.4685	−0.3597	−0.3616	−0.3572	1nn
bacc	CE	HierCE	True	0.4256	0.4685	−0.3880	−0.3910	−0.3853	1nn
bacc	HierSupCon	HierCE	True	0.4214	0.4848	−0.3847	−0.3871	−0.3816	20nn
bacc	CE	HierCE	True	0.4544	0.4848	−0.4159	−0.4159	−0.4130	20nn
bacc	HierSupCon	HierCE	True	0.4603	0.5144	−0.4222	−0.4240	−0.4203	logreg
bacc	SupCon	HierCE	True	0.5635	0.5144	−0.5255	−0.5278	−0.5228	logreg
	dino_L_HierCE	dino_L_Unsup	True	0.5144	0.5694	−0.4444	−0.4457	−0.4410	logreg
	dino_L_SupCon	dino_L_Unsup	True	0.5635	0.5694	−0.5253	−0.5280	−0.5226	logreg

Data availability

Codes are provided at https://github.com/mc2295/dino_hier.

References

Acevedo, A., Merino, A., Alférez, S., Molina, Á., Boldú, L., & Rodellar, J. (2020). A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in Brief, 30*, Article 105474.

Ahmed, K., Baig, M. H., Torresani, L. (2016). Network of experts for large-scale image categorization, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14, Springer, pp. 516–532.

Akata, Z., Reed, S., Walter, D., Lee, H., & Schiele, B. (2015). Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2927–2936).

Alipo-on, J., Escobar, F., Novia, J., Atienza, M., Mana-ay, S., Tan, M., AlDahoul, N., Yu, E., (2022). Dataset for machine learning-based classification of white blood cells of the juvenile visayan warty pig.

Aslan, M. A. (2020). Blood cell detection dataset. *Kaggle*.

Bertinetto, L., Mueller, R., Tertikas, K., Samangooei, S., & Lord, N. A. (2020). Making better mistakes: Leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12506–12515).

Cai, D., Chen, J., Zhao, J., Xue, Y., Yang, S., Yuan, W., Feng, M., Weng, H., Liu, S., Peng, Y., et al. (2024). Hicervix: An extensive hierarchical dataset and benchmark for cervical cytology classification. *IEEE Transactions on Medical Imaging*.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9650–9660).

Chen, R. J., Ding, T., Lu, M. Y., Williamson, D. F., Jaume, G., Chen, B., Zhang, A., Shao, D., Song, A. H., Shaban, M., et al. (2024). Towards a general-purpose foundation model for computational pathology. *Nature Medicine*. https://doi.org/10.1038/s41591-024-02857-3

Chomean, S., Khemtonglang, N., Mukda, E., & Kaset, C. (2025). Ai-powered body fluid cell classification: Development and validation using roboflow and yolov11n framework. *Telematics and Informatics Reports, 19*, Article 100243. https://doi.org/10.1016/j.teler.2025.100243

Diehl, A., Meehan, T., Bradford, Y., Brush, M., Dahdul, W., Dougall, D., & He, Y. (2016). The cell ontology 2016: Enhanced content, modularization and ontology interoperability. *Journal of Biomedical Semantics, 7*, 1–10.

Diehl, A. D., Meehan, T. F., Bradford, Y. M., Brush, M. H., Dahdul, W. M., Dougall, D. S., He, Y., Osumi-Sutherland, D., Ruttenberg, A., Sarntivijai, S., et al. (2016). The cell ontology 2016: Enhanced content, modularization, and ontology interoperability. *Journal of Biomedical Semantics, 7*, 1–10.

Ding, T., Wagner, S. J., Song, A. H., Chen, R. J., Lu, M. Y., Zhang, A., Vaidya, A. J., Jaume, G., Shaban, M., Kim, A., et al. (2024). Multimodal whole slide foundation model for pathology, arXiv preprint arXiv:2411.19666.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., & Mikolov, T. (2013). Devise: A deep visual-semantic embedding model. *Advances in Neural Information Processing Systems, 26*.

Hehr, M., Sadafi, A., Matek, C., Lienemann, P., Pohlkamp, C., Haferlach, T., Spiekermann, K., & Marr, C. (2023). Explainable ai identifies diagnostic cells of genetic aml subtypes. *PLOS Digital Health, 2*(3), Article e0000187.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D., (2020). Supervised contrastive learning, in: Advances in Neural Information Processing Systems, Vol. 33, pp. 18661–18673.

Koch, V., Wagner, S. J., Kazeminia, S., Sancar, E., Hehr, M., Schnabel, J. A., Peng, T., & Marr, C. (2024). Dinobloom: A foundation model for generalizable cell embeddings in hematology. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 520–530).

Koohbanani, N. A., Jahanifar, M., Tajadin, N. Z., & Rajpoot, N. (2020). Nuclick: A deep learning framework for interactive segmentation of microscopic images. *Medical Image Analysis, 65*, Article 101771.

Kouzehkanan, Z. M., Saghari, S., Tavakoli, E., Rostami, P., Abaszadeh, M., Mirzadeh, F., Satlsar, E. S., Gheidishahran, M., Gorgi, F., Mohammadi, S., et al., (2021). Raabin-wbc: a large free access dataset of white blood cells from normal peripheral blood, bioRxiv 2021–05.

Kouzehkanan, Z. M., Saghari, S., Tavakoli, S., Rostami, P., Abaszadeh, M., Mirzadeh, F., Satlsar, E. S., Gheidishahran, M., Gorgi, F., Mohammadi, S., et al. (2022). A large dataset of white blood cells containing cell locations and types, along with segmented nuclei and cytoplasm. *Scientific Reports, 12*(1), 1123.

Matek, C., Schwarz, S., Spiekermann, K., Marr, C., (2019). Human-level recognition of blast cells in acute myeloid leukaemia with convolutional neural networks, Nature Machine Intelligence 1 (11), 538–544, number: 11 Publisher: Nature Publishing Group. doi:10.1038/s42256-0190101-9. URL https://www.nature.com/articles/s42256-019-0101-9.

Matek, C., Krappe, S., Münzenmayer, C., Haferlach, T., & Marr, C. (2021). Highly accurate differentiation of bone marrow cell morphologies using deep neural networks on a large image data set. *Blood, 138*(20), 1917–1927. https://doi.org/10.1182/blood.2020010568

Muller, B., & Smith, W. (2020). A hierarchical loss for semantic segmentation. *Proceeding of VISAPP/VISIGRAPP*.

Naruenatthanaset, K., Chalidabhongse, T. H., Palasuwan, D., Anantrasirichai, N., Palasuwan, A. (2020). Red blood cell segmentation with overlapping cell separation and classification on imbalanced dataset, arXiv preprint arXiv:2012.01321.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. (2023). Dinov2: Learning robust visual features without supervision, arXiv preprint arXiv:2304.07193.

Rezatofighi, S. H., & Soltanian-Zadeh, H. (2011). Automatic recognition of five types of white blood cells in peripheral blood. *Computerized Medical Imaging and Graphics, 35*(4), 333–343.

Rieger, M. A., & Schroeder, T. (2012). Hematopoiesis. *Cold Spring Harbor Perspectives in Biology, 4*(12), Article a008250.

Shahzad, M., Ali, F., Shirazi, S. H., Rasheed, A., Ahmad, A., Shah, B., & Kwak, D. (2024). Blood cell image segmentation and classification: A systematic review. *PeerJ Computer Science, 10*, e1813.

Shetab Boushehri, S., Gruber, A., Kazeminia, S., Matek, C., Spiekermann, K., Pohlkamp, C., Haferlach, T., Marr, C. (2025). A large expert-annotated single-cell peripheral blood dataset for hematological disease diagnostics, medRxiv 2025–02.

Sonar, S. C., & Bhagat, K. (2015). An efficient technique for white blood cells nuclei automatic segmentation. *International Journal of Scientific & Engineering Research, 6*(5), 172–178.

Zhang, S., Xu, R., Xiong, C., & Ramaiah, C. (2022). Use all the labels: A hierarchical multi-label contrastive learning framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 16660–16669).

Zheng, X., Wang, Y., Wang, G., & Liu, J. (2018). Fast and robust segmentation of white blood cell images by self-supervised learning. *Micron, 107*, 55–71.

Zhou, J., Wei, C., Wang, H., W., Shen, Xie, C., Yuille, A., Kong, T. (2021). ibot: Image bert pretraining with online tokenizer, arXiv preprint arXiv:2111.07832.

Zhu, X., Bain, M. (2017). B-cnn: branch convolutional neural network for hierarchical classification, arXiv preprint arXiv:1709.09890.