Automatic Embedding Interventions for the Classification of Hematopoietic Cells

Philipp Gräbel
Institute of Imaging and Computer Vision
RWTH Aachen University, Germany
Email: graebel@lfb.rwth-aachen.de

Julian Thull
Institute of Imaging and Computer Vision
RWTH Aachen University, Germany

Martina Crysandt
Dept. of Hematology and Oncology
University Hospital
RWTH Aachen University, Germany

Barbara M. Klinkhammer
Institute of Pathology
University Hospital
RWTH Aachen University, Germany

Peter Boor
Institute of Pathology
University Hospital
RWTH Aachen University, Germany

Tim H. Brümmendorf
Dept. of Hematology and Oncology
University Hospital
RWTH Aachen University, Germany

Dorit Merhof
Institute of Imaging and Computer Vision
RWTH Aachen University, Germany

***Abstract*—The classification of hematopoietic cells is the most essential step in automating the analysis of human bone marrow samples. However, the complex structure of cell classes as well as class imbalance make this a challenging task, even for neural networks. Based on projective latent interventions, we propose automatic interventions that iteratively update a learned embedding with suitable transformations that shift different cell types apart and contract samples of the same type together. We present different ways of applying these: either directly on a higher-dimensional embedding or in a parametric version in two dimensions. We analyze the hyper-parameters and evaluate the proposed approach on a challenging dataset of hematopoietic cells. The results show an improvement of up to 3 percentage points for the classification F-score.**

I. Introduction

Many hematopoietic diseases are diagnosed based on cell count statistics in bone marrow samples. In particular, many types of leukemia can be recognized by a shift from a typical to an abnormal distribution of cell types. In clinical practice, this distribution is determined by medical specialists, who count the occurrences of individual cell types in microscopy images of a bone marrow sample. This, however, restricts the number of cells that can be realistically counted per patient.

An automated classification of cell types in bone marrow microscopy images is a key step in supporting hematological experts in diagnosis and research. Neural networks, which are often employed in similar scenarios, are capable of processing a much larger number of cell images in shorter time. If sufficiently trained, such a classifier could consequently yield more objective and reproducible estimates of the cell distribution as a basis for a more reliable diagnosis.

The core challenges for such classification tasks lie in the difficulty of hematopoietic data. First, there is a large number of different cell types. Second, some cells have large inter-class variabilities but are visually similar to other cell types. Third, the normal distribution of cell types results in high class imbalance. Furthermore, relationships between cell types exist, as blood cells mature in the bone marrow in a continuous process within different lineages.

For the classification of hematopoietic cells, both classical pipelines based on feature extractors and simple classifiers, and deep learning approaches have been investigated [1]. Of these, deep learning approaches have clearly proven more successful. Particularly, architectures such as *DenseNet* [2] have shown superior results [3]. In further research, alternative classifiers (e.g. VGG-net [4]) and the detection of hematopoietic cells (e.g. using R-CNN Networks [5], [6], [7]) have been investigated [8], [9], [10].

It has been shown that representation learning and dedicated techniques for improving the embedding can help with the problem of class imbalance, for example with the Class-center Triplet Loss [11]. Hinterreiter et al. [12] published the idea of manually interacting with learned embeddings in order to achieve a more suitably arranged embedding space for classification tasks. They propose to increase the distance between samples of adjacent classes and decrease the spread of embeddings of samples of a single class through manual interventions. These manual interventions, denoted as *Projective Latent Interventions* (PLIs), are performed on a two-dimensional visualisation of the latent space: a human-in-the-loop selects class clusters, which are shifted away from each other and contracted to the cluster center. In order to enable this human interaction, the embedding is transformed using a parametric dimensionality reduction, which can also be used in loss computation. This method shows improved classification results when used in conjunction with a typical classification loss. The disadvantages of PLIs lie in the required human

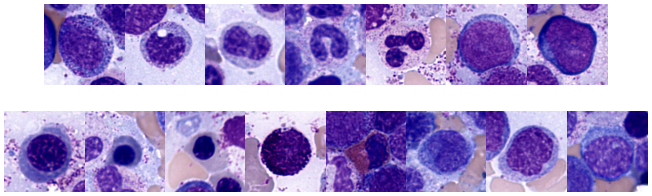

Fig. 1. Top: promyelocyte, myelocyte, metamyelocyte, band granulocyte, segmented granulocyte, blast cell, proerythroblast. Bottom: basophilic, polychromatic and orthochromatic erythroblast, basophilic and an eosinophilic granulocyte, promonocyte, monocyte, and lymphocyte.

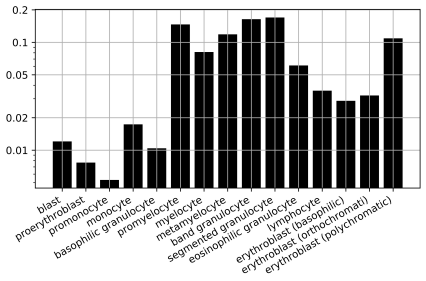

Fig. 2. Distribution of cell types as a logarithmic bar plot.

interaction as well as the reduction to only two dimensions.

To address these limitations, we propose *Automatic Latent Interventions* (ALI). In this approach, we replace the human-in-the-loop with a heuristic that automatically determines suitable clusters and performs interventions on them. This allows for multiple interventions during the training process without manual supervision and can work on arbitrary embedding sizes. We propose several suitable types of interventions as well as criteria and metrics for their application. Furthermore, we propose a parametric version that performs interventions in 2D. We show that these methods greatly improve results compared to PLIs as well as triplet margin loss (TML).

II. Image Data

In this work, we utilize a dataset of hematopoietic cell images from human bone marrow samples. Each sample is preprocessed using the established and well-defined Pappenheim staining procedure [13]. The digitization is performed using a whole slide image scanner with a magnification of $63\times$ and automatic immersion oiling. From each whole slide image representative regions are extracted in accordance with the typical workflow in hematological diagnosis.

The positions of individual cells are determined by a U-Net [14] combined with the Watershed algorithm [15] and are then manually validated. For each cell, the cell type is assigned by medical experts according to the classes in Figure 1. Different cell types occur in very different numbers in the human bone marrow and, therefore, in this dataset. Figure 2 shows the distribution of samples per cell type and highlights the class imbalance. For the classification task, we have a total of 4560 individual cell images patches of size $224 \times 224\,\mathrm{px}$.

Based on the description of image acquisition and annotation process in publications using other datasets, the dataset in this work is more challenging. Compared to [1], the dataset includes images with higher variation in staining and visual appearance, making the classification task more difficult. The samples in this work were specifically selected to achieve a high but realistic variability with respect to staining and digitization. This makes it more challenging compared to [10], who instead selected images and regions that are specifically suitable for classification. Furthermore, we include a larger number of cell types compared to [9] for a more valuable clinical diagnosis. This dataset aims at a challenging but accurate representation of data from the clinical workflow.

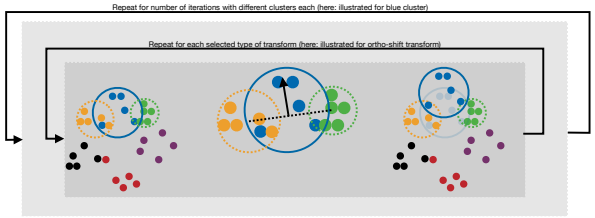

Fig. 3. Illustration of the intervention workflow performed after each training epoch to determine a transformed embedding, which is employed to determine the loss for the next training epoch. Here, the blue cluster is identified by the choice criterion, with green and orange as the two nearest neighbors (left). Based on the ortho-shift transform, a transformation of the sample embeddings of the blue cluster is determined (middle). Applying this yields the transformed embedding (right). In this example, only a single intervention iteration with only a single transform (ortho-shift) are shown. Typically, multiple types of interventions are applied over multiple iterations (each for different clusters).

III. Methods

We propose to perform interventions on classes based on a criterion Ω. We propose several types of interventions as well as a parametric and a class-weighted version.

A. Automatic Latent Interventions (ALI)

Automatic Latent Interventions (ALI) aim at incrementally improving learned embeddings by performing a series of transformations on them. Each intervention is constructed to increase the inter-class distances or to decrease the intra-class distances of different clusters. Note that each cluster refers to the embeddings of objects with the same class.

After each epoch, we automatically select suitable class clusters, which may benefit from interventions. Based on a fixed number of neighboring clusters, we then perform interventions transforms – e.g., to shift the selected cluster further away from those neighboring clusters. This process is illustrated in Figure 3. The transformed embedding is then used in an additional loss for the next training epoch.

Formally, we learn class-wise affine transformations $\theta_c(x_c) = R_c x_c + T_c$ with a rotation R_c and a shift T_c to all embeddings $x_c \in X_c$ for a class c. These transformations are applied to the sample representations of the clusters, thereby modifying their relative positions and spread.

Let B be a batch of representations and let $x_i^{'} = \theta_c(x_i)$ for all $x_i \in B$ of class c. Our proposed intervention loss is then defined as

$$\mathcal{L}^{\text{ALI}} = (1-\alpha)\mathcal{L}_{\text{tml}}(\mathcal{T}_B) + \frac{\alpha}{|B|}\sum_{x_i \in B}\mathcal{L}_1(x_i, x_i') \quad (1)$$

with $\mathcal{L}_{\text{tml}}$ the triplet margin loss, $\mathcal{T}_B$ all mined triplets in the batch that violate the margin, and $\mathcal{L}_1$ the distance between two vectors. The factor α weighs the Triplet Margin Loss (TML) and the ALI-loss. We choose the L1 distance to measure the loss as it is less affected by the "curse of dimensionality" compared to other distance norms such as MSE.

We construct θ_c as a concatenation of multiple interventions for each cluster $\theta_c = \theta_c^{n-1} \circ \theta_c^{n-2} \circ ... \circ \theta_c^0$. In each of the n iterations, one class cluster c is chosen using a criterion Ω as described in the following section. The nearest neighboring clusters are then used to construct the transformations θ_c^i such that they increase the distances of cluster c to its closest neighboring clusters and reduce its spread.

The class-wise transformation parameters R_c and T_c are updated after each applied intervention. The learned transformation for cluster c in iteration i is parameterized by R_c^i, T_c^i. The update is computed as $R_c = R_c^i R_c$ and $T_c = R_c^i T_c + T_c^i$.

B. Choice Criterion Ω

In order to decide which cluster to update next, we define a criterion Ω. From a metric learning standpoint, it is most beneficial to further separate a cluster c from the closest neighboring cluster, i.e. $\arg\min_i\{d_{i,j} \quad | \forall i \neq j\}$, where $d_{i,j}$ is the L2-distance between the centers of clusters i and j.

Instead of a deterministic criterion, it may be advantageous to employ a *probabilistic criterion*. To do so, we define different logits terms $l_i^{\text{criterion}}$:

- $l_i^{\text{mean-std}} = \text{std}(X_i)/\min_j d_{i,j}$
- $l_i^{\text{mean-all}} = d_{\text{all},i}^{-1}$

with $d_{\text{all},i}$ the average distance of cluster i to all other clusters and $\text{std}(X_i)$ the standard deviation of all embeddings belonging to the cluster i.

These terms are multiplied per class and normalized, yielding a probability distribution P over the classes. We sample from P in each iteration to select the next cluster $\Omega \sim P$.

C. Types of Interventions

We use contractions, shifts and rotations as possible interventions. Contraction and shift transforms have been proposed for PLI, with parameters determined by the human-in-the-loop during training. Based on a visual analysis of overlapping clusters in the two-dimensional embedding space, we further propose the rotation transform and an orthogonal shift transform. All of these are applied successively in each iteration i: $\theta_c^i = \theta_c^{\text{shift}} \circ \theta_c^{\text{rotate}} \circ \theta_c^{\text{contract}}$.

We define the *contraction* function as $\theta_c^{\text{contract}}(x) = x + \lambda_c(\bar{c} - x)$. Each data point x is pulled towards its cluster mean $\bar{c}$ using a scale factor λ_c, thereby effectively reducing the intra-class distances for this cluster.

The *rotation* function is defined as $\theta_c^{\text{rotate}}(x) = R_c(\lambda_r)x$, where $R_c(\lambda_r)$ represents the rotation matrix depending on a parameter λ_r for class c. We use it to rotate cluster c away from its neighboring clusters. λ_r quantifies the angle to rotate one cluster c_1 away from another c_2 (0 maps c_1 onto c_1 whilst 1 maps c_1 onto $-c_2$):

$$R_c(\lambda_r) = \prod_{k \in NN(c)} R_{ck}(\lambda_r) \quad \forall c \in M \quad (2)$$

In order to obtain R_{ij}, we compute the normal vectors that span the plane defined by v_i & v_j' with $v_j' = v_j - n_i \cdot v_j \times n_i$ and $n_x = \frac{v_x}{\|v_x\|}$. The rotation matrix can then be computed as

$$R_{ij}(\lambda_r) = I \qquad + (n_j' n_i^T - n_i n_j'^T)\sin(\alpha(\lambda_r)) \quad (3)$$

$$+ (n_i n_i^T + n_j' n_j'^T)(\cos(\alpha(\lambda_r)) - 1)\,. \quad (4)$$

The angle $\alpha(\lambda_r)$ is based on λ_r as described above.

The *shift* function is defined as $\theta_c^{\text{shift}}(x) = x + \lambda_s v_c$, where v_c represents the shift vector for class c that is used to shift it away from neighboring clusters using a scale factor λ_s. The direction of v_c is determined by a linear combination of two different shift vectors v_{shift} and v_{ortho}.

v_{shift} moves a cluster c away from neighboring clusters. To do so, we require the vectors from the cluster center $\bar{c}$ to the center of the neighboring clusters. These are weighted by their inverse distances and summed up, yielding v_{shift}.

v_{ortho} is based on vectors towards the cluster center $\bar{c}$, which are orthogonal to the lines that connect neighboring cluster centers as illustrated in Figure 3. These are weighted by their inverse distances and summed up, yielding v_{ortho}.

The linear combination $v_{\text{both}} = \gamma v_{\text{shift}} + (1-\gamma)v_{\text{ortho}}$ is used as the direction of the shift vector v_c based on the weight $\gamma \in [0, 1]$, which is an additional hyper-parameter. The magnitude is set to the scaled standard deviation $\lambda_s \text{std}(X_c)$ of the embeddings X_c of cluster c such that $v_c = \lambda_s \text{std}(X_c)\frac{v_{\text{both}}}{||v_{\text{both}}||}$. In order to keep the embedding norm approximately constant over the iterations, we normalize the shift vectors for all clusters after each iteration such that the norm of the original and transformed embedding remain equal.

D. Parametric Automatic Latent Interventions (PALI)

We also perform interventions in a lower dimensional parametric space, similar to [12]. This approach represents a less restrictive form of guidance as the network is not forced to use certain embedding positions but only needs to recreate the intervened neighborhood relations. The loss function uses parametric t-SNE (PTSNE) and is given by:

$$\mathcal{L}^{\text{PALI}} = (1-\alpha)\mathcal{L}_{tml}(\mathcal{T}_B) + \frac{\alpha}{|B|}\sum_{x_i \in B}\mathcal{L}_1(\text{PTSNE}(x_i), x_i')\,. \quad (5)$$

E. Class Imbalance

To mitigate the problem of class imbalance in the dataset, we weigh the original and intervened embeddings with class-wise factors before computing the L1 distance. The weight factor for class c with N_c samples in a dataset with N samples distributed across $|M|$ classes is computed as inverse class-frequency $w_c = \frac{N}{|M| * N_c}$.

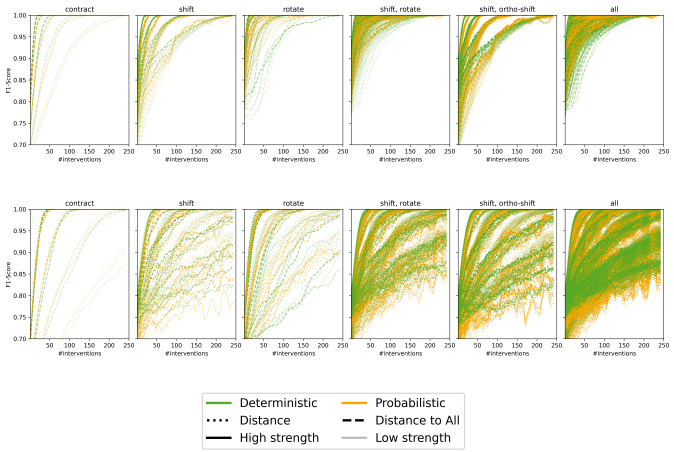

Fig. 4. Results of the hyper-parameter analysis for ALI (top) and PALI (bottom). Each plot shows one combination of intervention types. All of them contain the contraction transform even if not specified. For the sake of presentation, only the plots corresponding to the subsequently evaluated intervention types are shown even though all possible combinations have been tested. Each plot shows the macro F1-score from classification on the intervened embedding. The x-axis represents how often each intervention transform has been applied. The color denotes the choice criterion (deterministic or probabilistic), the line style whether only the inverse distance to the neighboring clusters or additionally the distance to all clusters is used. "Strength" refers to the λ parameters – high strength implies high λ-values.

F. Experimental Setup

We use a DenseNet-121 [2] architecture, which is pre-trained on the ImageNet dataset [16]. This network architecture has been shown as the most successful for a wide range of hyper-parameters in other works [3]. Augmentations include dropout ($p = 0.05$), random crops (11 px offset) and full random rotation.

The embedding is obtained through a final fully-connected layer with $n = 256$ neurons subsequent to the DenseNet. Also in the case of PALI, an embedding of this length is computed, even though its dimensionality is reduced through paramatric t-SNE for the computation of interventions.

Classification is performed with an SVM [17] using an RBF kernel [17] with the embedding vector as input. The corresponding hyper-parameters are optimized in grid search for every five epochs.

We employ 6-fold cross-validation with one fold each for testing and validation. The split is defined in such a way that each fold has similar distributions of cell types. Nevertheless, it is ensured that cell image patches from the same region of the whole slide image are not spread across multiple folds. Training is stopped early if the F-score on the validation set does not improve for 50 epochs.

1) Baselines: Due to the nature of other datasets (see Section II), a direct comparison to results from the corresponding works is not possible. To make the proposed methods comparable to baseline experiments, we perform the two referenced methods on the same dataset. First, we use TML [18], which is the standard technique in representation learning. Second, we compare to PLI with manually chosen interventions.

2) Hyper-parameter Experiment: In order to evaluate configurations and hyper-parameters of ALI, we perform a dry-run analysis on typical embeddings obtained from training with TML [18]. We iteratively apply interventions to this embedding, train a classifier on the training sets and evaluate it on the validation set. This is performed for ALI as well as PALI in 6-fold cross-validation. As this requires no network training, a large number of parameters can be evaluated.

In a first dry-run, we evaluate the following parameters: $\lambda_{\text{contr}} = 2^x$, $x \in [-4, -1]$, $\lambda_{\text{rot}} = 0.01 \cdot 2^x$, $x \in [1, 5]$, $\lambda_{\text{shift}} = 2^x$, $x \in [-2, 3]$, $\gamma \in [0, 0.25, 0.5, 0.75, 1]$ and all proposed criteria Ω taking 15 neighbors into account. We then choose the scale parameters (λ, γ) for each combination of interventions separately, and determine Ω jointly. In a second dry-run, we evaluate the number of neighbors (between 1 and 15) for these parameters.

3) Intervention Experiment: For the final experiments, the following interventions are evaluated (each additionally including contraction): contraction only, shift, rotation, shift & rotation, shift & ortho-shift, and all types of interventions. The evaluation covers Automatic Latent Interventions in n dimensions (nD, ALI) with $n = 256$ as well as the combination of TML and ALI in 2D with $\alpha = 0.9$ as well as n dimensions with $\alpha = 0.8$. These values for α were identified as suitable parameters in a preliminary hyper-parameter optimization step. In the 2D case, Parametric ALI (PALI) are used. Furthermore, we evaluate all methods with and without applying class-weights (CW) to the loss. Interventions are recomputed and applied at the beginning of every epoch.

IV. Results

A. Hyper-parameters

The dry-run experiments are used to determine the most suitable hyper-parameter combinations. The success of each run is based on the maximum classification score and the number of iterations needed to reach this score. This is shown in Figure 4, which shows one curve per combination of hyper-parameters for each intervention type combination.

In all cases involving the contraction intervention, a final score of 1.0 can be reached in usually less than 15 iterations. Without contraction, the final scores are typically lower. Consequently, further experiments focus only on combinations of intervention types that include contraction (even if not explicitly mentioned).

In the clear majority of cases, a deterministic criterion is the most beneficial: the argmax of the product of *mean-std* and *mean-all* is the best choice for Ω in the n-dimensional case and only *mean-std* for two dimensions. These criteria are used for the remainder of the experiments.

For each combination of intervention types, the other hyper-parameters, such as scale parameters (λ, γ), are set to the most successful combination (cf. Table I).

B. Interventions

Figure 5 shows the resulting macro F-scores for the baselines (TML only and PLIs [12]) as well as the proposed

TABLE I

RESULTING PARAMETERS FOR THE EVALUATED COMBINATIONS OF CONTRACT (C), SHIFT (S), ORTHO-SHIFT (OS), AND ROTATION (R). n IS THE NUMBER OF NEIGHBORS, i THE NUMBER OF REQUIRED INTERACTIONS. PARAMETERS ARE GIVEN AS (2D, ND). $\lambda_C = 0.5$.

	λ_r	λ_s	γ	n	i
c	-, -	-, -	-, -	15, 15	48, 33
c, s	-, -	1, 4	-, -	11, 2	48, 25
c, r	.16, .32	-, -	-, -	11, 11	47, 15
c, s, os	-, -	2, 4	.75, .5	14, 2	46, 25
c, s, r	.02, .32	1, .25	-, -	15, 15	45, 15
all	.02, .32	1, 1	.25, .75	15, 15	43, 15

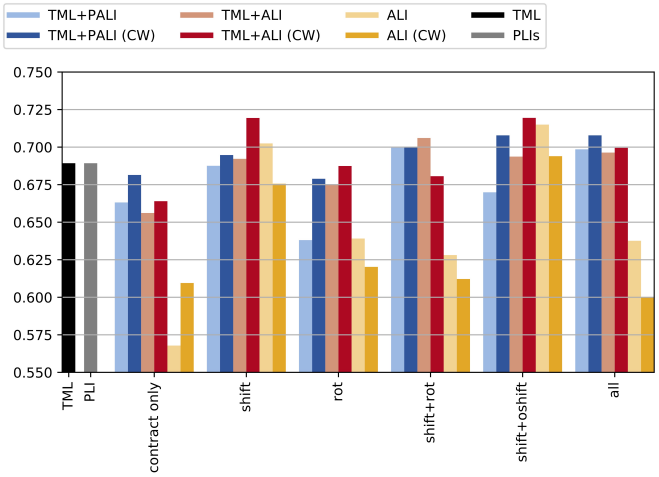

Fig. 5. Classification results in terms of macro F1 score. The baselines are Triplet Margin Loss (TML, black) and Projective Latent Interventions (PLI, grey). For each type of intervention (differing in the selected types of transforms), a group of six bars shows the resulting scores. The color denotes the loss method: PALI (blue) or ALI (red) as described and ALI without the TML loss ($\alpha = 0$, orange). For each of these, the shade denotes whether class weights are applied (darker) or not (lighter) for each loss.

methods. In most cases, ALI+TML achieves better scores than PALI or ALI without TML. Compared to both baselines (PLIs and TML only), shift interventions in particular improve the classification score, whereas rotation as well as contraction only are not beneficial. Generally, the best results are achieved with contraction and shift (either normal shift or normal+ortho shift) together with TML and class weights. This results in a macro F-score improvement from 0.689 to 0.719.

Table 2 lists the results in tabular form, supplementing all standard deviations over the cross validation folds.

V. DISCUSSION

The proposed experimental setup can be used to automatically evaluate a large number of hyper-parameters without network training. Consequently, this can be performed with low computational cost for each new use-case. Additionally, the results of the hyper-parameter search show many combinations leading to good results in the dry-run. This indicates a stability towards hyper-parameters, which needs to be evaluated when utilized in the training process in future work.

The optimization of hyper-parameters and particularly Figure 4 also yield interesting insights into the design of the

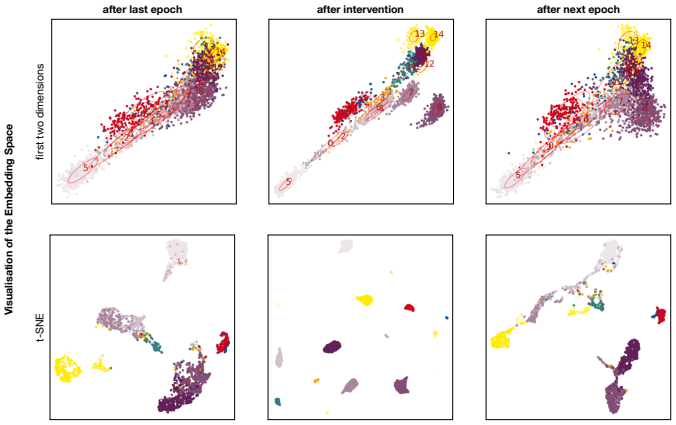

Fig. 6. Visualisation of the embedding space (top: first two dimensions, bottom: supervised t-SNE) before (left) and after interventions (middle) as well as after another epoch based on the interventions (right). This visualisation is after the third (left, middle) and fourth (right) epoch of training with ALI using contraction and shifts. In this case, the parameters are larger than usual for illustrative purposes. Each color denotes a different cell type.

method. It should first be noted that high parameter strength (i.e., high values for the λ-parameters) usually yield better results in the dry-run experiment. For the contraction-transform in particular, this is no suprise, as it results indirectly in loss terms that "pull" embeddings closer to the cluster center. For other transforms, this is not necessarily the case.

The plots of the hyper-parameter optimization show a tendency to form almost discrete groups of curves. This often indicates a stronger dependency on the intervention strength compared to the choice criterion: often, similar strength values result in similar curves independent of the choice criterion. This is particularly the case for individual transforms.

Nevertheless, tendencies regarding the suitability of choice criteria can be made: the deterministic criterion typically outperforms the probabilistic ones. This indicates that improving the lower performing classes is more important than generally optimizing the embedding space. This fact also gives an indication of the success of the proposed methods: the transforms optimize the embedding class-wise without being direclty influenced by the number of samples, which reduces the impact of class imbalance. These findings can also be exploited in future research, focusing on techniques to intervene even more strongly with underperforming classes – for example, based on the confusion between classes.

In general, PALI transforms have a slower convergence but also a larger spread of results. Whereas with ALI, almost every combination of hyper-parameters led to good results (albeit slower) in the dry-run experiments, PALI is more sensitive to them and yields lower results for less optimal parameters.

Figure 6 visualizes how interventions can impact the embedding space. It shows that the intervened embedding (middle) has a much clearer separation of clusters due to the shift and contraction transforms. Employing this embedding in the ALI loss function for another epoch, improves the embedding space even further. Particularly the shift of certain clusters is visible

TABLE II
TABULAR CLASSIFICATION RESULTS IN TERMS OF MACRO F1 SCORE AND STANDARD DEVIATION OVER ALL FOLDS IN PERCENT. THE BASELINE SCORED 68.9 ± 2.87 (TML) AND 68.9 ± 2.15 (PLI).

intervention	TML+PALI	TML+PALI (CW)	TML+ALI	TML+ALI (CW)	ALI	ALI (CW)
contract only	66.3 ± 3.46	68.1 ± 2.44	65.6 ± 3.29	66.4 ± 2.00	56.8 ± 3.91	61.0 ± 1.42
shift	68.8 ± 1.81	69.5 ± 1.27	69.2 ± 1.67	71.9 ± 2.21	70.2 ± 1.71	67.6 ± 3.63
rot	63.8 ± 1.24	67.9 ± 3.85	67.5 ± 1.20	68.7 ± 2.51	63.9 ± 2.78	62.0 ± 2.18
shift+rot	70.0 ± 1.87	70.0 ± 2.26	70.6 ± 1.23	68.1 ± 0.97	62.8 ± 2.11	61.2 ± 1.94
shift+oshift	67.0 ± 1.79	70.8 ± 1.49	69.4 ± 0.89	71.9 ± 1.95	71.5 ± 1.80	69.4 ± 1.30
all	69.9 ± 2.22	70.8 ± 2.35	69.6 ± 2.51	70.0 ± 3.23	63.8 ± 4.10	60.0 ± 2.98

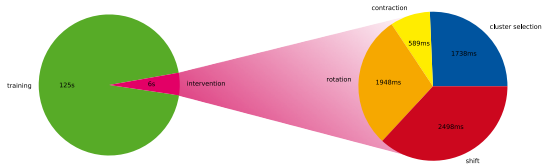

Fig. 7. Average training time per epoch on a *Core i5* CPU for the interventions and a 6 GB *GTX 1060* GPU for network training.

(e.g., with the yellow clusters 13 and 14, or the purple clusters 7 and 8, which no longer overlap after the intervention). Also, the contraction is noticeable to a smaller extent (e.g., with cluster 5). One advantage of ALI is that interventions are performed multiple times throughout the training process. This allows certain transforms to have more or less impact on the embedding space depending on its current state in the training.

The hyper-parameter experiment gives no direct insight into which transforms perform best. Nevertheless, the results indicate that the contraction transform is necessary. Consequently, we analyse a wide range of interventions that include this transform in the next evaluation.

The final training results clearly show that automatic interventions improve classification scores in our representation learning scenario. For the classification of hematopoietic cells, an F-score improvement of 3 percentage points can be achieved. Whereas rotation-based interventions show low performance in most cases, particularly for pure ALI loss without TML, combinations of contractions and shifts yield excellent results. In the case of contraction with normal (non-orthogonal) shifts, applying a class weight to the loss improves results even more. Even for parametric interventions in lower dimensions, improvements can be observed.

Table II shows the precise F-score results as well as the standard deviation across the cross-validation folds. It can be noticed that the contract only intervention experiments experience high F-score variations, likely because contracted clusters optimize targets only in terms of intra-class distance. By proposing contracted representations, the network is discouraged from finding new cluster positions with better inter-class distances. This makes the cluster positions highly dependent on the model initialization. The combination with the TML can remedy this to some extent, as noticable in the comparision of standard deviations for ALI with TML+ALI. The usage of all intervention types together leads to similar large spreads as the representations are altered so severely that the learning process becomes unstable. The overall lowest F-score deviations are achieved for the shift and shift+oshift interventions, which also achieve excellent F-scores.

Interventions are only performed in network training, not in the prediction phase. Thus, all computational overheads only prolong the training time but have no influence on the runtime for the actual application of the network. Figure 7 shows more details regarding the computation time of individual steps. It shows that, compared to the training time, the computation for interventions is very fast – even when employing all intervention types. Shifting takes longest, as it is composed of two individual transforms (normal and orthogonal shifts).

The proposed methods do not incorporate domain knowledge, thus their implementation is straightforward for other use-cases as well. A focus for further research is the success on other datasets as well as the choice of hyper-parameters in those cases.

VI. CONCLUSION

This paper presents the novel idea of automatic interventions to further improve the learned embedding in representation learning tasks. This includes multiple types of intervention (contraction, shift, ortho-shift, rotation) and different ways of application (parametric, combination with triplet loss, class-weights). We evaluate this on a challenging medical dataset of hematopoietic cells. The results show that automatic latent interventions can improve the classification score on learned embeddings beyond what is possible with the triplet loss.

ACKNOWLEDGMENT

This work is a purely retrospective, pseudonymized analysis of bone marrow samples under the Helsinki Declaration of 1975/2000 with written informed consent of all patients. This work was funded by the Federal Ministry of Education and Research (BMBF) and the Ministry of Culture and Science of North Rhine-Westphalia as part of the National High-Performance Computing four Computational Engineering Sciences (NHR4CES) consortium. This work was further supported by the German Research Foundation (DFG; Project IDs: 322900939, 454024652 and 445703531), European Research Council (ERC, Consolidator Grant No 101001791), Federal Ministry of Health (Deep Liver, No. ZMVI1-2520DAT111), and the Federal Ministry of Economic Affairs and Energy (EMPAIA, No. 01MK2002A). The authors would like to thank Reinhild Herwartz and Melanie Baumann for their efforts in sample preparation and annotation.

REFERENCES

[1] P. Gräbel, M. Crysandt, R. Herwartz, M. Hoffmann, B. M. Klinkhammer, P. Boor, T. H. Brümmendorf, and D. Merhof, "Evaluating out-of-the-box methods for the classification of hematopoietic cells in images of stained bone marrow," in *1st MICCAI Workshop on Computational Pathology (COMPAY)*, 2018.

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[3] P. Gräbel, G. Nickel, M. Crysandt, R. Herwartz, M. Hoffmann, B. M. Klinkhammer, P. Boor, T. H. Brümmendorf, and D. Merhof, "Systematic analysis and automated search of hyper-parameters for cell classifier training," in *IEEE International Symposium on Biomedical Imaging (ISBI)*, 2020.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[8] P. Gräbel, Ö. Özkan, M. Crysandt, R. Herwartz, M. Hoffmann, B. M. Klinkhammer, P. Boor, T. H. Brümmendorf, and D. Merhof, "Circular anchors for the detection of hematopoietic cells using retinanet," in *IEEE International Symposium on Biomedical Imaging (ISBI)*, 2020.

[9] J. W. Choi, Y. Ku, B. W. Yoo, J.-A. Kim, D. S. Lee, Y. J. Chai, H.-J. Kong, and H. C. Kim, "White blood cell differential count of maturation stages in bone marrow smear using dual-stage convolutional neural networks," *PLOS ONE*, vol. 12, no. 12, p. e0189259, Dec. 2017.

[10] R. Chandradevan, A. A. Aljudi, B. R. Drumheller, N. Kunananthaseelan, M. Amgad, D. A. Gutman, L. A. D. Cooper, and D. L. Jaye, "Machine-based detection and classification for bone marrow aspirate differential counts: initial development focusing on nonneoplastic cells," *Laboratory Investigation*, vol. 100, no. 1, pp. 98–109, Sep. 2019.

[11] W. Lei, R. Zhang, Y. Yang, R. Wang, and W.-S. Zheng, "Class-center involved triplet loss for skin disease classification on imbalanced data," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 1–5.

[12] A. Hinterreiter, M. Streit, and B. Kainz, "Projective latent interventions for understanding and fine-tuning classifiers," in *Interpretable and Annotation-Efficient Learning for Medical Image Computing*. Springer, 2020, pp. 13–22.

[13] T. Binder, H. Diem, R. Fuchs, K. Gutensohn, and T. Nebe, "Pappenheim stain: Description of a hematological standard stain – history, chemistry, procedure, artifacts and problem solutions," *Journal of Laboratory Medicine*, vol. 36, no. 5, pp. 293–309, 2012.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[15] S. Beucher and F. Meyer, "The morphological approach to segmentation: the watershed transformation," *Mathematical morphology in image processing*, vol. 34, pp. 433–481, 1993.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[17] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[18] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.