Diffeomorphism Matching for Fast Unsupervised Pretraining on Radiographs

Thanh M. Huynh*[1]
v.thanhhuynh@vinbrain.net

Chanh D. T. Nguyen*[1, 2]
v.chanhndt@vinbrain.net

Ta Duc Huy[1]
v.huyta@vinbrain.net

Hoang Cao Huyen[1]
v.huyenhc@vinbrain.net

Trung H. Bui[3]
bhtrung@yahoo.com

Steven QH Truong[1]
brain01@vinbrain.net

[1] VinBrain JSC
7 Bang Lang, Viet Hung District
Ha Noi, Vietnam

[2] VinUniversity
Vinhomes Ocean Park, Gia Lam District
Ha Noi, Vietnam

[3] Independant Researcher
USA

Abstract

Unsupervised pretraining is an approach that leverages a large unlabeled data pool to learn data features. However, it requires billion-scale datasets and a month-long training time to surpass its supervised counterpart on fine-tuning in many computer vision tasks. In this study, we propose a novel method, Diffeomorphism Matching (DM), to overcome those challenges. The proposed method combines self-supervised learning and knowledge distillation to equivalently map the feature space of a student model to that of a big pretrained teacher model. On the Chest X-ray dataset, our method alleviates the need to acquire billions of radiographs and substantially reduces pretraining time by 95%. In addition, our pretrained model outperforms other pretrained models by at least 4.2% in F1 score on the CheXpert dataset and 0.7% in Dice score on the SIIM Pneumothorax dataset. Code and pretrained model are available at `https://github.com/jokingbear/DM.git`

1 Introduction

Deep Learning (DL), in particular, supervised learning, generally requires large-scale and high-quality labeled datasets to achieve human-level of accuracy. However, obtaining these datasets is costly and time-consuming in the medical domain. This is due to the expertise required to label a large amount of data, and doctors' consensus cannot be reached easily. In order to avoid those issues, transfer learning from a pretrained model was adopted to train a high-performance model without a massive amount of available labeled data. Several approaches in medical image analysis showed that using the ImageNet pretrained models substantially outperform their trained from scratch counterpart on many tasks [19].

[*] Equal Contribution.

Recently, unsupervised pretraining approaches remove the barrier of having annotation in pretext tasks. In MoCo [4], the authors showed that unsupervised pretraining a model on a billion-scale dataset outperforms its supervised counterpart. However, acquiring such a large dataset in medical imaging is impractical due to complicated paperwork procedures and costs. Furthermore, current approaches to unsupervised pretraining require lots of computational resources [3, 7, 8]. This paper proposes a novel method, Diffeomorphism Matching (DM), to tackle those issues. Our motivation is based on the equivalence of feature spaces between a big pretrained teacher model and a much smaller randomly initialized student model. The teacher model is pretrained on the billion-scale natural image dataset IG-1B [20]. We use the language of differential geometry to model the equivalence between feature spaces. Our method reduces the need to acquire a billion-scale radiograph dataset. Experiments show that our pretrained model outperforms other supervised and unsupervised pretrained models in downstream tasks such as classification and segmentation. Compared to other unsupervised pretraining methods, ours requires less computational resources to train and reduce training duration by 20 times. In summary, our contributions are as followed:

- We proposed a loss function that makes the feature space of a randomly initialized student model equivalent to that of a teacher model using the language of differential geometry.

- Our proposed loss combined with distillation loss function and a big pretrained model results in an unsupervised pretraining procedure that is faster, more stable, and requires less computation resource than contemporary unsupervised pretraining approaches.

- We conducted extensive experiments on Chest X-ray datasets for classification and segmentation tasks. Compared to other pretrained models, ours outperform them in fine-tuning. This increment in fine-tuning performance shows that our approaches reduce the need to acquire a billion-scale dataset in Chest X-ray imaging.

2 Related Works

Contrastive Learning. Instead of training a classification model to extract features in pretext tasks, Contrastive Learning compares different views of the same image to extract invariant features without any usage of annotation. In order to compare images, current approaches leverage Siamese networks to extract features from two different views of images to form a pair. A positive pair consists of two different views of the same image, and a negative pair consists of views of different images. The objective is to increase feature similarity between positive pairs and decrease the similarity for negative pairs. In doing so, the model learns to extract view-invariant and discriminative features of the input data. In SimCLR [3], the authors proposed using the contrastive loss with large batch size, strong data augmentations, and long training time to achieve a good pretrained model. MoCo 1 and 2 [4, 8] use a running dictionary that gets updated after each backward propagation step to reduce the batch size while keeping a big negative pool for contrastive loss. In BYOL [7], the authors only compare positive pairs. In order to avoid collapsing to a trivial solution, they remove the gradient of a feature in a pair to create an asymmetry in the similarity objective function. SimCLR, MoCo, and BYOL achieved comparable performances on transferring to downstream tasks compared with supervised pretraining when trained on ImageNet [6] dataset.

Knowledge Distillation. Our approach is inspired, in part, by knowledge distillation. REFILLED [31] treats models as two parts, the embedding and the top-layer classifier. The teacher first distills its knowledge to the student by bridging the gap between non-overlapping label spaces in the top-layer classifier. In the second stage, the local embedding centers of the teacher further improve training for the student. In Factor Transfer [15], the authors proposed using convolutional modules as paraphrasers and s for knowledge transfer between teacher and student model. The paraphraser and the translator extract teacher and student factors respectively to calculate the factor transfer loss, which is then minimized during training. Jing *et al.* [30] uses adaptive instance normalization to transfer the learned feature statistics back to the teacher to determine whether such statistics learned by the student are reliable. Recently, Li *et al.* [17] proposed using a student model with the same architecture as the teacher but without residual paths. The student model's outputs at each resolution stage are the inputs for the teacher's next resolution stage. The whole pipeline is trained using classification loss and final layer feature matching loss. The approach allows gradient to flow through the teacher model, which can be very computationally expensive. Therefore, the method has limited application to big model regimes. In addition to matching feature maps at the same resolution, Chen *et al.* [2] proposed using feature maps at different resolutions as an additional guide to the distillation process. However, the feature matching process can be computationally expensive for a big teacher model due to having different resolution feature maps for each resolution.

3 Method

3.1 Motivation

To learn features from a big pretrained model, we first define the equivalence between networks. Our definition is motivated by [23]. However, the definition in [23] is restrictive since it only deals with the same output dimensions architectures. To overcome such restriction, we create a more generalized definition using the language of differential geometry. Let M be a manifold consists of all data points of a dataset, a Convolutional Neural Network (CNN) F with Batch Normalization [12] and skip connections can be treated as a C^1 mapping from M to some feature manifold $F(M)$. It is possible to treat F as a C^1 mapping because of the Lipschitz condition enforcing of Batch Normalization [24] and empirically smoothing of skip connection [18] present in all of the standard CNN architectures. We also treat $F(x)$ as a local coordinate of $F(M)$ for each data point $x \in M$. Then the new equivalent definition can be stated as follows.

Definition 1 *CNN T and CNN S are equivalent when $\forall\ x,y \in M$, $S(x) = S(y)$ if and only if $T(x) = T(y)$.*

Definition in [23] is a special case of ***Def. 1*** because for any 2 data points x, y, we have $T(x) = S(x)$ and $T(y) = S(y)$, hence when T extracts the same feature on x and y, S will extract the same feature on those data points. Moreover, ***Def. 1*** has a nice mathematical property when T and S are constant rank mappings, there exists a manifold structural equivalent mapping between $T(M)$ and $S(M)$, i.e., a diffeomorphism.

Theorem 1 ([16]) *If T and S are constant rank C^1 mappings from M to some feature manifolds $T(M)$ and $S(M)$, such that $\forall x,y \in M$ $T(x) = T(y)$ if and only if $S(x) = S(y)$, then there exists a diffeomorphism between $T(M)$ and $S(M)$.*

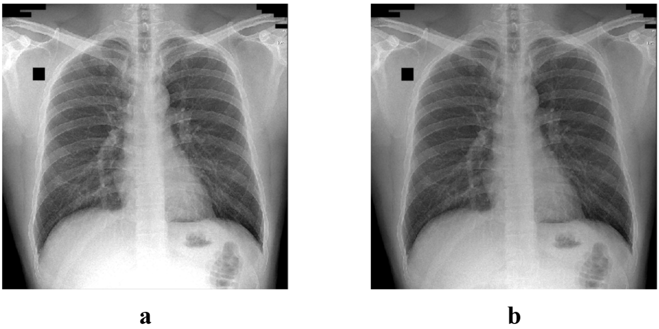

Figure 1: A chest X-ray image and its slight variation in brightness and contrast version. The features are extracted using the pretrained ResNext-32x48d-WSL [20]. Their features Euclidean distance is approximately 5×10^{-3} which is very closed considering the high dimensionality of the feature space. **a**. Original Image. **b.** The same image with small variation in brightness and contrast.

3.2 Diffeomorphism Matching

Def. 1 may not hold precisely in practice due to the high dimensionality of the extracted features, i.e., 2 data points with the same features are likely to be the same. In addition to that, The requirement of constant rank mapping for a CNN may not hold because of the black-box nature of deep learning models. However, Fig. 1 shows that an image and its slight variation in brightness and contrast version can have features that are very closed to each other (Euclidean distance on order of 10^{-3}). We treat this case as approximately equal in practice. Furthermore, as can be shown, to make two networks equivalent as in ***Def. 1***, one only needs to make sure that their image is equivalent and their mapping diagram is commutative (see Fig. 2).

$$\begin{tikzcd}
 & M \arrow[dl, "T"'] \arrow[dr, "S"] & \\
T(M) \arrow[rr, shift right, "H^{-1}"'] & & S(M) \arrow[ll, shift right, "H"']
\end{tikzcd}$$

Figure 2: Mapping diagram of 2 CNNs, T and S are equivalent CNN, H and H^{-1} are their diffeomorphic mapping and its inverse.

Theorem 2 *if T and S are C^1 mapping from data manifold M to some feature manifolds $T(M)$ and $S(M)$ such that there exists a diffeomorphism H with the property $H \circ S = T$ and $H^{-1} \circ T = S$, then T and S are equivalent (proof in supplementary material).*

Based on Theorem 2, we propose a loss function that consists of 2 parts: commutative and identity.

$$\mathcal{L}_{\text{DM}} = \mathcal{L}_{\text{c}} + \mathcal{L}_{\text{id}}, \tag{1}$$

where $\mathcal{L}_c$ is for ensuring that Fig. 2 is commutative, and $\mathcal{L}_{id}$ is to constraint the mapping H between $S(M)$ and $T(M)$ to be diffeomorphic. More specifically,

$$\mathcal{L}_c = \sum_{x \in D} \|HS(x) - T(x)\|_2^2 + \|H^{-1}T(x) - S(x)\|_2^2, \tag{2}$$

$$\mathcal{L}_{id} = \sum_{x \in D} \|H^{-1}HS(x) - S(x)\|_2^2 + \|HH^{-1}T(x) - T(x)\|_2^2, \tag{3}$$

where D is the training dataset, T is a pretrained teacher CNN, S is a randomly initialized student CNN, and H, H^{-1} are modeled by two 3-layers fully connected networks. Since a diffeomorphism mapping is, at the very least, a C^1 mapping, we use the smooth activation function ELU [5] in both H and H^{-1} to constraint them to have continuous derivative, hence making it easier to approximate a diffeomorphism. Furthermore, as shown in [34], low-level features learned by pretrained networks contain meaning such as corner and edge extraction. Therefore, we add a feature transfer loss $\mathcal{L}_{FT}$ [9] to guide the student model to learn meaningful low-level representation.

$$\mathcal{L}_{\text{FT}} = \sum_i \begin{cases} 0 & \text{if } r(S'_i) \leq T'_i \leq 0. \\ (r(S'_i) - T'_i)^2 & \text{otherwise.} \end{cases} \tag{4}$$

where i runs over all of feature map width, height, and channel, S' and T' are the last feature maps at each resolution of S and T respectively, and r is a 1x1 Convolution layer that transforms the feature S' to match the dimension of T'. $\mathcal{L}_{\text{FT}}$ weakly matches the feature map T' and S' by aligning positive regions and negative regions where S' is bigger than T'. More details description of $\mathcal{L}_{\text{FT}}$ can be found in [9]. The final loss function is

$$\mathcal{L} = \mathcal{L}_{\text{DM}} + \lambda \mathcal{L}_{\text{FT}}. \tag{5}$$

Fig. 3 shows the overview of our training process. S and T models' intermediate features are matched together using $\mathcal{L}_{\text{FT}}$ while their final global features are matched together using $\mathcal{L}_{\text{DM}}$. During the training process, there's no gradient flowing back to the T model.

4 Experimental Result

4.1 Implementation Details

We leverage both public and private chest X-ray datasets in pretraining. For public datasets, we make use of CheXpert [13], MIMIC-CXR [14], PadChest [1], and ChestX-ray14 [28]. For private dataset, we collected chest X-ray images from local hospitals (Due to anonymity, we will release the data collection location later). The combined dataset contains approximately 1.2M unlabeled radiographs used for self-supervised pretraining. Table 1 shows the detail of these datasets.

After getting the full dataset, we resize all radiographs to have a min size of 320 while keeping the aspect ratio. The images are then normalized to have intensity values in the range $[-1, 1]$. We use random crop to 224×224, horizontal flip, random brightness, translation, rotation (max $\pm 35°$), and scale for augmentations. Finally, we add cutout with a maximum of 3 patches of 32x32 pixels.

We use billion-scale dataset pretrained model ResNext101-32x48d-WSL [20] for T model to learn rich pretrained features. We also leverage the RegnetY design space in [21] to search

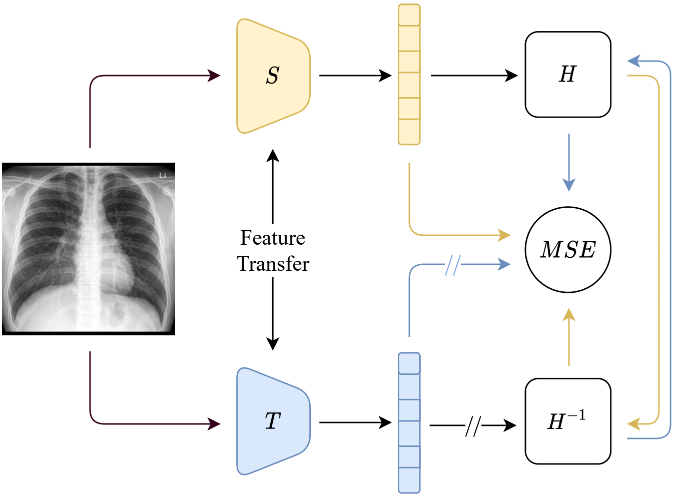

Figure 3: Overview of DM training procedure. Same color arrows are matched together in MSE loss. The slash on the arrows show no gradient is propagated back to the T model.

Table 1: Datasets used in pretraining. #Label is the number of labeled radiograph. #Total is the dataset size. DM pretraining does not use label.

Dataset	#Label	#Total	Label Usage in Pretraining
CheXpert	223,648	223,648	No
MIMIC-CXR	257,989	257,989	No
PadChest	154,396	154,396	No
ChestX-ray14	112,120	112,120	No
Private Dataset	130,030	484,235	No
Final Dataset	-	1,232,388	No

for a good S model for chest X-ray tasks. Our design spaces consist of width multiplier $w_m \in [2, 2.5]$, bottleneck block ratio $b \in [0.25, 1]$, and SE [10] squeeze ratio $s \in [1/16, 1/2]$. We trained 2K models sampled from all the design spaces on CheXpert dataset and analyze the final result using EDF [21]. The best design space has $w_m = 2.5$, $b = 1$, and $s = 1/16$ (more details in supplement). The final model, called RegChest, has approximately 63M total number of parameters. For Eq. 5, we use $\lambda = 1$ in our training pipeline. We follow the same setup as [9] for $\mathcal{L}_{\text{FT}}$.

For optimizer, we use SGD with Nesterov momentum of 0.9. The optimal initial learning rate was found to be 1.5, using the approach in [26]. We adopt the one-cycle (Super Convergence) learning rate scheduling approach in [26] to train RegChest for 10 epochs. We use the trained S model for fine-tuning downstream tasks. The whole training process was on 4 NVIDIA v100 GPUs with a batch size of 128 (32 each GPU).

4.2 Fine-tuning

We compare our pretrained model against standard pretrained models on two downstream tasks, classification and segmentation. To reduce the effect of randomness, we trained each model 3 times and took the average and standard deviation performance. For standard pretrained models, we used Densenet121 [11], ResNext101-32x8d [29], RegNetY-16GF [21], Efficientnet-B7 [27], ResNext101-32x8d-WSL [20], and ResNext101-32x16d-WSL [20]. For the classification task, we fine-tuned pretrained models on CheXpert 13 abnormal findings for 8 epochs using one-cycle scheduler (Super Convergence) [26] for fast convergence with an input size of 512x512. The training procedure is kept the same for all the backbones. Due to its massive size, we didn't fine-tune the teacher model ResNext-32x48d-WSL because we can't replicate the training condition of other models on it. Table 2 shows AUC scores of all pretrained models on 500 radiographs in CheXpert test set.

DM based pretrained model RegChest peaks the best natural image pretrained model (Densenet121) on AUC score by 0.4%. However, due to the imbalance between positive and negative instances for each finding, AUC score is not the best performance indicator. We, therefore, measure the F1 score of each model on CheXpert test set. Our pretrained model surpassed the best natural image pretrained model (ResNext101-32x8d) on 9/13 findings with an average F1 score gain of 4.2% (Table 2).

Table 2: AUC and F1 scores of pretrained models on all findings on CheXpert test set. $O(10^{-4})$ stands for value smaller than 5×10^{-4}. The best results are in bold. The average AUC gain of DM and the best ImageNet pretrained model is 0.4%. The average F1 score gain of DM and the best ImageNet pretrained model is 4.2%.

Findings	**Densenet121**	**Resnext101** **32x8d**	**Resnext101** **32x16d-WSL**	**Efficientnet** **B7**	**RegnetY** **16GF**	**RegChest** **DM**
			AUC			
Average	0.898 ± 0.006	0.885 ± 0.003	0.887 ± 0.004	0.875 ± 0.005	0.883 ± 0.005	**0.902 ± 0.005**
Enlarged Cardiomediastinum	**0.808 ± 0.006**	0.793 ± 0.006	0.783 ± 0.005	0.751 ± 0.011	0.760 ± 0.002	0.798 ± 0.007
Cardiomegaly	**0.856 ± 0.003**	0.843 ± 0.003	0.838 ± 0.002	0.806 ± 0.004	0.815 ± 0.006	**0.856 ± 0.003**
Lung Opacity	**0.944 ± 0.004**	0.943 ± $O(10^{-4})$	0.933 ± $O(10^{-4})$	0.934 ± $O(10^{-4})$	0.937 ± 0.001	0.940 ± 0.002
Lung Lesion	0.961 ± 0.019	0.961 ± 0.019	0.926 ± 0.003	0.926 ± 0.003	0.934 ± $O(10^{-4})$	**0.979 ± 0.002**
Edema	0.896 ± 0.002	0.899 ± 0.001	0.895 ± 0.005	0.900 ± 0.003	0.896 ± 0.003	**0.906 ± $O(10^{-4})$**
Consolidation	0.776 ± 0.009	0.761 ± 0.005	0.763 ± 0.003	0.765 ± 0.007	0.778 ± 0.002	**0.787 ± 0.012**
Pneumonia	0.794 ± 0.009	0.805 ± 0.002	0.795 ± 0.010	0.802 ± 0.004	**0.827 ± 0.009**	0.809 ± 0.006
Atelectasis	0.810 ± 0.008	0.811 ± 0.007	0.808 ± 0.001	0.807 ± 0.006	0.804 ± 0.003	**0.832 ± 0.004**
Pneumothorax	**0.992 ± 0.001**	0.991 ± $O(10^{-4})$	0.989 ± 0.001	0.989 ± 0.001	0.990 ± 0.002	**0.992 ± $O(10^{-4})$**
Pleural Effusion	0.962 ± 0.001	**0.966 ± $O(10^{-4})$**	0.959 ± $O(10^{-4})$	0.950 ± 0.001	0.958 ± 0.001	0.956 ± 0.001
Pleural Other	0.977 ± 0.007	0.965 ± 0.003	0.978 ± 0.004	0.955 ± 0.007	0.951 ± 0.009	**0.982 ± 0.008**
Fracture	**0.925 ± 0.006**	0.829 ± 0.011	0.894 ± 0.017	0.864 ± 0.016	0.864 ± 0.023	0.923 ± 0.015
Support Devices	**0.976 ± 0.001**	0.972 ± 0.002	0.974 ± 0.002	0.961 ± 0.001	0.971 ± $O(10^{-4})$	0.970 ± 0.001
			F1			
Average	0.532 ± 0.016	0.537 ± 0.008	0.523 ± 0.010	0.510 ± 0.022	0.516 ± 0.015	**0.579 ± 0.022**
Enlarged Cardiomediastinum	0.599 ± 0.031	0.578 ± 0.002	0.566 ± 0.012	0.571 ± 0.013	0.564 ± 0.014	**0.640 ± 0.009**
Cardiomegaly	**0.643 ± 0.019**	0.612 ± 0.005	0.604 ± 0.001	0.624 ± 0.012	0.602 ± 0.011	0.630 ± 0.009
Lung Opacity	0.878 ± 0.011	**0.879 ± $O(10^{-4})$**	0.862 ± 0.007	0.867 ± 0.007	0.870 ± 0.006	0.870 ± 0.011
Lung Lesion	0.427 ± 0.039	**0.547 ± 0.035**	0.483 ± 0.026	0.320 ± 0.015	0.415 ± 0.042	0.545 ± 0.025
Edema	0.610 ± 0.022	0.628 ± 0.002	0.614 ± 0.002	**0.642 ± 0.006**	0.637 ± 0.014	0.638 ± 0.005
Consolidation	0.320 ± 0.003	0.281 ± 0.006	0.282 ± 0.005	0.297 ± 0.020	0.333 ± 0.010	**0.352 ± 0.035**
Pneumonia	0.219 ± 0.019	0.230 ± 0.004	0.211 ± 0.009	0.188 ± 0.009	0.184 ± 0.009	**0.273 ± 0.018**
Atelectasis	0.587 ± 0.007	0.578 ± 0.001	0.583 ± 0.002	0.598 ± 0.012	0.598 ± 0.009	**0.672 ± 0.015**
Pneumothorax	0.563 ± 0.015	0.563 ± 0.025	0.546 ± 0.023	0.536 ± 0.051	0.502 ± 0.044	**0.596 ± 0.035**
Pleural Effusion	0.790 ± 0.014	0.789 ± 0.008	0.773 ± 0.010	0.767 ± 0.004	0.779 ± 0.005	**0.796 ± 0.005**
Pleural Other	0.218 ± 0.015	0.215 ± 0.011	0.221 ± 0.030	0.162 ± 0.136	0.154 ± 0.006	**0.347 ± 0.105**
Fracture	0.179 ± 0.011	0.197 ± 0.005	0.161 ± 0.002	0.180 ± 0.003	0.182 ± 0.018	**0.276 ± 0.002**
Support Devices	0.885 ± 0.007	0.890 ± 0.006	0.890 ± 0.008	0.874 ± 0.001	0.884 ± 0.008	**0.892 ± 0.005**

We also investigate whether our pretrained model transfers well to other downstream tasks such as segmentation. We use the SIIM Pneumothorax dataset [25], which consists of

10,675 radiographs with 8,296 negative and 2,379 positive data points for training and a test set of 1,372 radiographs with 1,082 negatives and 290 positives. For segmentation model, we adopt an Unet like architecture with pretrained models as backbones and an input size of 256x256. We train the models for 70 epochs using Adam optimizer with learning rate drop linearly from 10^{-3} to 0. The negatives proportion is gradually increased from 20% to 60% throughout training to address data imbalance. All pretrained backbones use the same training pipeline. As shown in Table 3, pretrained RegChest has a performance gain of 0.7% in Dice score compared with the best natural image pretrained model (Densenet121 and ResNext 32x8d-WSL). The competitive performances on CheXpert and SIIM Pneumothorax dataset show that features pretrained by DM transfer well to downstream tasks.

Table 3: Dice scores of pretrained models on SIIM Pneumothorax test set. The best Dice scores are in bold. RegChest DM has a gain of 0.7% compared with other pretrained model.

	Densenet121	**Resnext101 32x8d**	**Resnext101 32x8d-WSL**	**Efficientnet B7**	**RegnetY 16GF**	**RegChest DM**
Dice Score	0.808 ± 0.009	0.803 ± 0.009	0.808 ± 0.011	0.805 ± 0.004	0.806 ± 0.007	**0.815** ± **0.002**

4.3 Ablation Study

4.3.1 Loss Components and architecture

We study the contribution of each component in Eq. 5 as well as training RegChest from scratch. For the contribution of each term, $\mathcal{L}_{\text{DM}}$ outperforms $\mathcal{L}_{\text{FT}}$ by 2.2% on F1 score in classification and 0.7% on Dice score in segmentation (Table 4). The performance gains show that $\mathcal{L}_{\text{DM}}$ term in Eq. 5 contributes more than the $\mathcal{L}_{\text{FT}}$ term. The reason is that $\mathcal{L}_{\text{FT}}$ only weakly transfers feature maps, as shown in Eq. 4, while $\mathcal{L}_{\text{DM}}$ directly matches the final feature vector of T and S with each other; hence it has better learning signals. Together, they match local features and global features, leading to the best performance. For network architecture, Table. 4 shows that DM pretraining model significantly outperforms its training from scratch counterpart by 4.1% in AUC score and 6.6% in F1 score for classification, and 10.3% in Dice score for segmentation. Therefore, the performance gain of DM in Table. 2 and Table. 3 are from pretraining, not from better neural architecture.

Table 4: Contribution of each component of the loss function. $\mathcal{L}_{\text{DM}} + \mathcal{L}_{\text{FT}}$ in Eq. 5 achieves the best performance compared with training from scratch and each of its components. All methods use the same RegChest architecture.

No Pretrain	$\mathcal{L}_{FT}$	$\mathcal{L}_{DM}$	AUC (CheXpert)	F1 (CheXpert)	Dice (SIIM Pneumothorax)
✓	-	-	0.861 ± 0.034	0.513 ± 0.065	0.712 ± 0.009
-	✓	-	0.886 ± 0.003	0.515 ± 0.008	0.793 ± 0.004
-	-	✓	0.886 ± 0.004	0.537 ± 0.012	0.801 ± 0.003
-	✓	✓	**0.902** ± **0.005**	**0.579** ± **0.022**	**0.815** ± **0.002**

4.3.2 Compare with unsupervised pretraining

We compare DM with unsupervised pretraining methods that can be trained with 4 GPUs such as MoCo v2 [4] and C2L [35]. We use the official implementation of MoCo v2 and C2L on RegChest architecture. Due to the increase in memory size, we reduce the queue size from 65K to 32.7K. Because MoCo and C2L training are unstable initially, we were not able to apply one-cycle scheduling to it. Therefore, we follow the original setting to train 800 epochs for MoCo and 150 epochs for C2L with batch size 128. MoCo and C2L training procedures use the same hardware settings and dataset as DM.

Table 5: Fine-tune performance of MoCo v2, C2L and DM on CheXpert and SIIM Pneumothorax dataset. DM outperforms MoCo v2 and C2L in fine-tuning performance while substantially reducing training time.

Pretraining Methods	AUC (CheXpert)	F1 (CheXpert)	Dice (SIIM Pneumothorax)	Training Time (days)
MoCo v2	0.882 ± 0.004	0.533 ± 0.012	0.806 ± 0.007	28.30
C2L	0.889 ± 0.003	0.522 ± 0.009	0.805 ± 0.004	25.80
DM	**0.902** ± **0.005**	**0.579** ± **0.022**	**0.815** ± **0.002**	**1.35**

As shown in Table 5, DM outperforms MoCo v2 and C2L on downstream tasks with more than 1.3% gain in AUC score and 4.5% in F1 score on CheXpert, and 0.9% in Dice score on SIIM Pneumothorax while reducing the training duration by 95% (20 times faster). The substantial reduction in training time is due to a more stable loss function, leading to a more straightforward application of sophisticated learning rate scheduling. We hypothesize the F1, AUC, and Dice score performance gain is due to learning richer features from a big model trained on a billion-scale dataset. We acknowledge that using a pretrained teacher model is unfair to MoCo and C2L. However, integrating a big pretrained teacher model to the other methods is not trivial; hence it does not lie in the scope of this paper.

4.3.3 Compare with distillation method

Because our method can be treated as distillation from a teacher model to a student model, we compare our method with distillation methods that can be used for unsupervised pretraining such as Attention Transfer (AT) [33] , Feature Transfer (FT) [9] and Matching Guided Distillation (MGD) [32]. For AT, FT, and MGD, we only use their feature loss for pretraining.

Table 6: Performance of fine-tuned pretrained models using distillation methods and DM on CheXpert test set. DM surpassed the best distillation method with a 1.4% increase in AUC score and 3.7% increase in F1 score.

Methods	AUC	F1
AT	0.882 ± 0.006	0.542 ± 0.007
FT	0.886 ± 0.003	0.515 ± 0.008
MGD	0.893 ± 0.003	0.532 ± 0.007
DM	**0.902** ± **0.005**	**0.579** ± **0.022**

DM pretrained model outperforms AT, FT, and MGD on both AUC and F1 score on CheXpert with an average gain of 1.5% on AUC score and 4.9% on F1 score. (Table 6).

AT, FT, and MGD requires supervisory signal from a teacher model trained on labeled data to achieve a good final performance. Therefore, in the case of lacking such signal, their performances fall short compared to DM.

4.3.4 Different pretrained T models

We study the effect of pretraining the S model with different T models trained on ImageNet and IG-1B datasets. Table. 7 shows that using ResNext-32x8d-WSL T model improved F1 score of the S model by 1.5% on CheXpert test set compared to using the same T model pretrained on ImageNet dataset. The improvement indicates that using the T model pretrained on larger dataset results in a better S model. Furthermore, just like other distillation methods, the performance of the S model increases as the size of the T model increases. The increment in performance shows that bigger models extract more diverse features, which leads to better distillation results.

Table 7: Performance results of the S model using various T models on CheXpert test set. S achieves the best performance in AUC and F1 score when using the largest model on IG-1B dataset.

Pretrained T Model	# Parameters	AUC	F1
ResNext-32x8d	88M	0.878 ± 0.003	0.531 ± 0.013
ResNext-32x8d-WSL	88M	0.880 ± 0.003	0.546 ± 0.021
ResNext-32x16d-WSL	193M	0.886 ± 0.004	0.551 ± 0.020
ResNext-32x48d-WSL	829M	**0.902 ± 0.005**	**0.579 ± 0.022**

ImageNet pretrained model ResNext-32x8d performs almost the same as its student model in both AUC and F1 scores (Table 2 and Table. 7). This sameness shows that DM effectively preserves the performance of the T model. However, for the case of ResNext-32x16d-WSL model, Table. 7 shows that S improves the F1 score by 2.8% compared to T (Table. 2). We hypothesize this improvement is due to the over-parameterization [22] of the teacher model (193M parameters) compared with the student model (63M parameters). This over-parameterization leads to the teacher model being easily overfitted on the training dataset compared with the student model. Our method, DM, helps retain the teacher model's feature extraction capability while using a substantially smaller student model. We analyze the distribution of feature matching errors in the supplementary material.

5 Conclusion

This work proposed a Diffeomorphism Matching training method that uses pretrained features of a model trained on a billion-size dataset. By using such a large model trained on a big dataset, our final model surpasses other pretrained models on downstream tasks such as classification and segmentation. The improvement in performances on downstream tasks shows that our method alleviates the need to acquire a billion-size dataset in chest X-ray imaging. Furthermore, our method substantially decreases unsupervised pretrainining time compared with other methods. We hypothesize that integrating a pretrained teacher model into contrastive self-training loops would improve its performance on downstream tasks further. We leave it to future work to explore this approach.

References

[1] Aurelia Bustos, Antonio Pertusa, José María Salinas, and Maria de la Iglesia-Vayá. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Medical Image Anal.*, 66:101797, 2020. doi: 10.1016/j.media.2020.101797. URL https://doi.org/10.1016/j.media.2020.101797.

[2] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5008–5017. Computer Vision Foundation / IEEE, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Chen_Distilling_Knowledge_via_Knowledge_Review_CVPR_2021_paper.html.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020. URL http://proceedings.mlr.press/v119/chen20j.html.

[4] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020. URL https://arxiv.org/abs/2003.04297.

[5] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.07289.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://doi.org/10.1109/CVPR.2009.5206848.

[7] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html.

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June*

13-19, 2020, pages 9726–9735. IEEE, 2020. doi: 10.1109/CVPR42600.2020.00975. URL https://doi.org/10.1109/CVPR42600.2020.00975.

[9] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 1921–1930. IEEE, 2019. doi: 10.1109/ICCV.2019.00201. URL https://doi.org/10.1109/ICCV.2019.00201.

[10] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(8):2011–2023, 2020. doi: 10.1109/TPAMI.2019.2913372. URL https://doi.org/10.1109/TPAMI.2019.2913372.

[11] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.243. URL https://doi.org/10.1109/CVPR.2017.243.

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/ioffe15.html.

[13] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn L. Ball, Katie S. Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 590–597. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.3301590. URL https://doi.org/10.1609/aaai.v33i01.3301590.

[14] Alistair E. W. Johnson, Tom J. Pollard, Seth J. Berkowitz, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih-ying Deng, Roger G. Mark, and Steven Horng. MIMIC-CXR: A large publicly available database of labeled chest radiographs. *CoRR*, abs/1901.07042, 2019. URL http://arxiv.org/abs/1901.07042.

[15] Jangho Kim, Seonguk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/6d9cb7de5e8ac30bd5e8734bc96a35c1-Paper.pdf.

[16] John M. Lee. *Introduction to Smooth Manifolds - Chapter 5 - Theorem 5.21*. Springer New York, 2012. doi: 10.1007/978-1-4419-9982-5. URL https://doi.org/10.1007/978-1-4419-9982-5.

[17] Guilin Li, Junlei Zhang, Yunhe Wang, Chuanjian Liu, Matthias Tan, Yunfeng Lin, Wei Zhang, Jiashi Feng, and Tong Zhang. Residual distillation: Towards portable deep neural networks without shortcuts. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/657b96f0592803e25a4f07166fff289a-Abstract.html.

[18] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6391–6401, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/a41b3bb3e6b050b6c9067c67f663b915-Abstract.html.

[19] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A. W. M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Anal.*, 42:60–88, 2017. doi: 10.1016/j.media.2017.07.005. URL https://doi.org/10.1016/j.media.2017.07.005.

[20] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, volume 11206 of *Lecture Notes in Computer Science*, pages 185–201. Springer, 2018. doi: 10.1007/978-3-030-01216-8_12. URL https://doi.org/10.1007/978-3-030-01216-8_12.

[21] Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10425–10433. IEEE, 2020. doi: 10.1109/CVPR42600.2020.01044. URL https://doi.org/10.1109/CVPR42600.2020.01044.

[22] Maithra Raghu, Chiyuan Zhang, Jon M. Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3342–3352, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/eb1e78328c46506b46a4ac4a1e378b91-Abstract.html.

[23] David Rolnick and Konrad P. Kording. Reverse-engineering deep relu networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8178–8187. PMLR, 2020. URL http://proceedings.mlr.press/v119/rolnick20a.html.

[24] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2488–2498, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/905056c1ac1dad141560467e0a99e1cf-Abstract.html.

[25] SIIM. Siim pneumothorax segmentation dataset, 2019. data retrieved from Kaggle, https://www.kaggle.com/c/siim-acr-pneumothorax-segmentation.

[26] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of residual networks using large learning rates. *CoRR*, abs/1708.07120, 2017. URL http://arxiv.org/abs/1708.07120.

[27] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019. URL http://proceedings.mlr.press/v97/tan19a.html.

[28] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3462–3471. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.369. URL https://doi.org/10.1109/CVPR.2017.369.

[29] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5987–5995. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.634. URL https://doi.org/10.1109/CVPR.2017.634.

[30] Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via adaptive instance normalization, 2020.

[31] Han-Jia Ye, Su Lu, and De-Chuan Zhan. Distilling cross-task knowledge via relationship matching. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. doi: 10.1109/cvpr42600.2020.01241.

[32] Kaiyu Yue, Jiangfan Deng, and Feng Zhou. Matching guided distillation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XV*, volume 12360 of *Lecture Notes in Computer Science*, pages 312–328. Springer, 2020. doi: 10.1007/978-3-030-58555-6_19. URL https://doi.org/10.1007/978-3-030-58555-6_19.

[33] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sks9_ajex.

[34] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer, 2014. doi: 10.1007/978-3-319-10590-1_53. URL https://doi.org/10.1007/978-3-319-10590-1_53.

[35] Hong-Yu Zhou, Shuang Yu, Cheng Bian, Yifan Hu, Kai Ma, and Yefeng Zheng. Comparing to learn: Surpassing imagenet pretraining on radiographs by comparing image representations. In Anne L. Martel, Purang Abolmaesumi, Danail Stoyanov, Diana Mateus, Maria A. Zuluaga, S. Kevin Zhou, Daniel Racoceanu, and Leo Joskowicz, editors, *Medical Image Computing and Computer Assisted Intervention - MICCAI 2020 - 23rd International Conference, Lima, Peru, October 4-8, 2020, Proceedings, Part I*, volume 12261 of *Lecture Notes in Computer Science*, pages 398–407. Springer, 2020. doi: 10.1007/978-3-030-59710-8_39. URL https://doi.org/10.1007/978-3-030-59710-8_39.