

Low-Rank Knowledge Decomposition for Medical Foundation Models

Yuhang Zhou^{1,2}, Haolin Li^{2,3}, Siyuan Du^{2,3}, Jiangchao Yao^{1,2,✉}, Ya Zhang^{1,2}, Yanfeng Wang^{1,2,✉}

¹Cooperative Medianet Innovation Center, Shanghai Jiao Tong University

²Shanghai Artificial Intelligence Laboratory ³Fudan University

{zhouyuhang, Sunarker, ya_zhang, wangyanfeng622}@sjtu.edu.cn,

{dusiyuan, lihaolin}@pjlab.org.cn

Abstract

The popularity of large-scale pre-training has promoted the development of medical foundation models. However, some studies have shown that although foundation models exhibit strong general feature extraction capabilities, their performance on specific tasks is still inferior to task-specific methods. In this paper, we explore a new perspective called “Knowledge Decomposition” to improve the performance on specific medical tasks, which deconstruct the foundation model into multiple lightweight expert models, each dedicated to a particular task, with the goal of improving specialization while concurrently mitigating resource expenditure. To accomplish the above objective, we design a novel framework named Low-Rank Knowledge Decomposition (LoRKD), which explicitly separates gradients by incorporating low-rank expert modules and the efficient knowledge separation convolution. Extensive experimental results demonstrate that the decomposed models perform well in terms of performance and transferability, even surpassing the original foundation models. Source code is available at: <https://github.com/MediaBrain-SJTU/LoRKD>

1. Introduction

Foundation models pre-trained on large-scale and diverse datasets, have been proven to possess powerful general feature extraction capabilities and can handle various tasks [4, 38]. However, some studies [15, 21, 49] have indicated that the performance of foundation models is still inferior to task-specific methods, suggesting that current foundation models are unable to simultaneously guarantee both generality and specialization. Moreover, with the gradual expansion of data scale and model capacity, the deployment costs of future foundation models may become exorbitant. To address them, we propose a new perspective called knowledge decomposition, aiming to offer potential solutions for the practical application of medical foundation models.

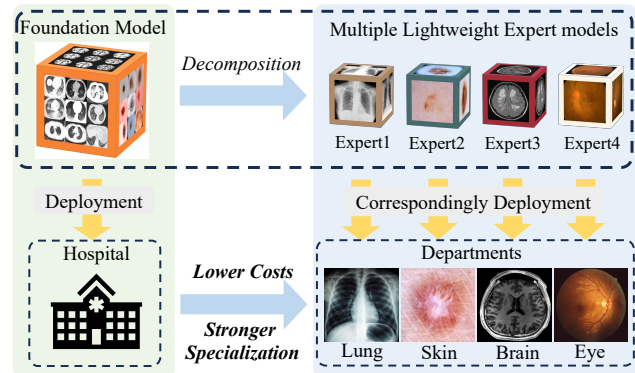


Figure 1. Knowledge decomposition is employed to break down the foundation model into multiple lightweight expert models, each dedicated to a specific domain. The goal of this paradigm is to improve the specialization of deployment models within a specific domain, while simultaneously reducing deployment costs.

The purpose of knowledge decomposition is to break down the foundation model into multiple lightweight expert models, where each expert model focuses solely on a specific domain, such as a department within a hospital (as shown in Figure 1). The resulting expert models, compared to the original foundation model, gain stronger specialization and lower deployment costs. In order to preserve both the performance and transferability of decomposed models, we need to inject task-specific knowledge and common knowledge into each expert model correspondingly, which is highly challenging. To the best of our knowledge, there has been no research conducted in the medical field on how to decompose a foundation model into multiple expert models. However, recently in the field of natural images, KF [52] has made preliminary explorations into this issue. KF decomposes the pre-trained model into a common knowledge network (CKN) and multiple task-specific networks (TSNs) by manipulating the mutual information between models. After decomposition, the CKN can be combined with each TSN to form task-specific expert models. However, in our experiments, we find that the effectiveness

of this method in medical scenarios is not significant.

In this paper, we propose a novel method for knowledge decomposition of medical foundation models called Low-Rank Knowledge Decomposition (LoRKD). LoRKD consists of two main components: low-rank expert modules and the efficient knowledge separation convolution. The former provides multiple parameter-efficient task-specific knowledge carriers for each convolution, which effectively controls the introduction of parameters while ensuring sufficient feature representation capability. The latter provides an efficient implementation method for expert knowledge separation at the convolutional level, allowing gradients to be separated into the corresponding expert modules in a single forward propagation, while accumulating them in the shared backbone. This ensures that each expert module learns task-specific knowledge while the shared backbone learns common knowledge. After decomposition, we can integrate the task-specific expert modules and the shared backbone through parameter fusion, ensuring model performance and transferability without increasing additional parameters. Furthermore, benefiting from the training pattern and parameter fusion mechanism, our decomposed model can easily switch task knowledge across different domains. The performance comparison on three pre-training datasets and seven downstream datasets demonstrates the effectiveness of LoRKD. A large number of analytical experiments further showcase the advantages of LoRKD from different perspectives. In a nutshell, our contributions can be summarized as the following:

- We introduce knowledge decomposition to broaden application of medical foundation models, which decomposes models into multiple lightweight experts to reduce costs and enhance specialization. The incorporation of this novel perspective offers potential solutions for the practical implementation of medical foundation models.
- We design a new method LoRKD, which consists of two components: low-rank expert modules and the efficient knowledge separation convolution. LoRKD injects task-specific knowledge into the corresponding expert modules through efficient explicit gradient separation.
- A significant number of experiments and analyses have demonstrated the effectiveness of LoRKD and the potential of knowledge decomposition.

2. Related work

Knowledge Distillation. Knowledge distillation (KD) [18] is an effective knowledge transfer method, which can be categorized into two groups: logits-based distillation [14, 31, 53] and feature-based distillation [3, 12, 33, 35, 42, 43, 46, 54]. The former encourages students to mimic the softmax outputs of teacher models, while the latter encourages students to mimic the intermediate-level features from the hidden layers of teacher models. Different from these methods

that focus on transferring complete knowledge, our goal is to decompose knowledge into different expert models.

Multi-Task Learning. Multi-task learning (MTL) [6, 10, 36] aims to train a unified model to solve multiple distinct but related tasks [7, 13, 25, 40, 41, 56]. Therefore, the main focus of MTL is to train better general feature extractor. For example, MoCo [13] addresses the convergence issue of Multi-Gradient Descent Algorithm (MGDA) [11] to ensure convergence to Pareto optima. Aligned-MTL [41] stabilizes the training process by aligning the principal components of the gradient matrix. However, tasks within the context of foundation models often display significant diversity, and solely pursuing common knowledge may not be appropriate. In summary, MTL focuses on extracting shared knowledge from relevant tasks, while knowledge decomposition emphasizes separating task-specific knowledge from the foundation model trained on diverse tasks.

Knowledge Decomposition. Different from the previous disentangled representation learning that are usually done through adversarial learning [9, 26, 29, 44] or variational auto-encoder [5, 17, 22], the goal of knowledge decomposition is to break down the pre-trained foundation model into multiple task-specific experts. Recently, in the field of natural images, KF [52] has explored knowledge decomposition by promoting modularization of knowledge through optimizing mutual information loss [19, 27, 32]. It decomposes a pre-trained model into a common knowledge network and multiple task-specific networks. In this paper, we conduct the first exploration of knowledge decomposition in the medical field and propose a novel approach that not only better controls the number of parameters but also attains a more advanced level of performance and transferability.

3. Proposed Method

Given a medical foundation model F_p pre-trained on a broad range of data, our goal is to decompose F_p into T lightweight expert models F_1, \dots, F_T that can be deployed to T different medical departments instead of using F_p . Our lightweight decomposition model comprises a shared backbone F_s and T expert modules $\{E_1, \dots, E_T\}$ during training. To achieve efficient knowledge decomposition, we propose low-rank expert modules and efficient knowledge separation convolution which will be described in detail below. An overview of our method can be seen in Figure 2.

3.1. Low Rank Expert Modules

Considering the limited computational resources and the scalability of the number of tasks, expert modules, as carriers of task-specific knowledge, need to strike a balance between the number of parameters and the feature representation capability. LoRA [20], as a commonly used fine-tune method in foundation models, has been proven to be parameter-efficient [47, 55]. Inspired by this, we propose to

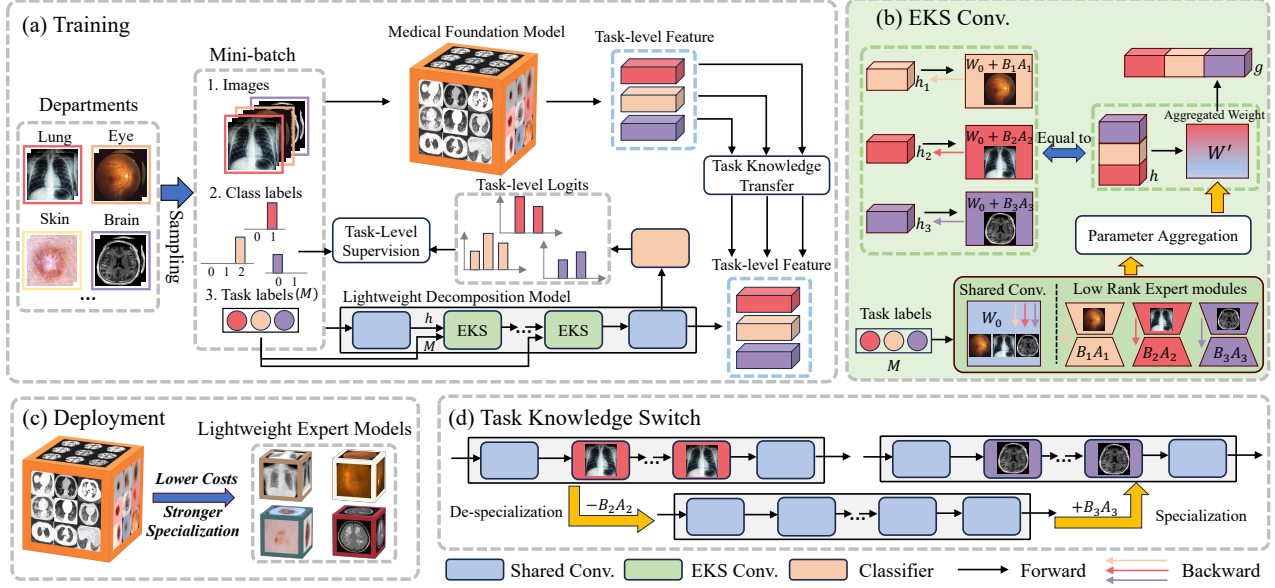


Figure 2. The overview of LoRKD. We introduce low-rank expert modules to control the number of parameters and efficient knowledge separation convolution to achieve computationally efficient explicit gradient separation. The decomposed models can replace the original foundation model in specific domains, and can switch task knowledge conveniently between different departments.

use a similar low-rank structure as the carriers for knowledge decomposition, named low rank expert modules.

Given a shared convolution $\mathbf{W}_0 \in \mathbb{R}^{C^{\text{out}} \times C^{\text{in}} \times k \times k}$ in F_s , where $C^{\text{out}}, C^{\text{in}}, k$ represent the number of output channels, the number of input channels, and the kernel size respectively, we configure two low-rank factors $\mathbf{B}_t \in \mathbb{R}^{C^{\text{out}} \times k \times r}$ and $\mathbf{A}_t \in \mathbb{R}^{r \times k \times C^{\text{in}}}$ for t -th expert, where r represents the rank. As a result, for the features belonging to the t -th task, original convolution operation $g_t = \mathbf{W}_0 h_t$ can be transformed into $g_t = (\mathbf{W}_0 + \mathbf{B}_t \mathbf{A}_t) h_t$, where, for brevity, we omit the reshape operation, and h_t, g_t represent the input features and output features respectively.

It is worth noting that, unlike previous scenarios where \mathbf{W}_0 remains fixed in LoRA, in our knowledge decomposition scenario, \mathbf{W}_0 , as a carrier of common knowledge, requires updating along with the low-rank factors.

3.2. Efficient Knowledge Separation Convolution

To achieve knowledge decomposition, we propose explicit gradient separation as our solution. This approach requires each expert module to compute gradients solely for its corresponding task, enabling the acquisition of task-specific knowledge. Simultaneously, the shared backbone collects gradients from all tasks to facilitate the acquisition of common knowledge among tasks. However, when a mini-batch of data contains T tasks, the convolution operation becomes T times $g_t = (\mathbf{W}_0 + \mathbf{B}_t \mathbf{A}_t) h_t$, where $t \in \{1, \dots, T\}$. The T times forward propagation significantly increases the training time, especially when a large number of tasks needed to be decomposed. To address this, we propose Ef-

icient Knowledge Separation Convolution (EKS Conv.).

In order to clarify our improvements in convolution, we first review the standard convolution operation. In a deep convolutional neural network, the input features of each convolution can be represented as $h \in \mathbb{R}^{B \times C^{\text{in}} \times H \times W}$, where B, H, W represent the sample number of a mini-batch, the height and width of the feature maps, respectively. If the kernel size of the convolution is k and the stride is 1, each output feature unit $o_{ij} \in \mathbb{R}^{B \times C^{\text{out}}}$ in output features $g \in \mathbb{R}^{B \times C^{\text{out}} \times H \times W}$ can be represented as

$$o_{ij} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} h_{(i+m)(j+n)} \cdot \omega_{mn},$$

where $i \in \{1, \dots, H\}, j \in \{1, \dots, W\}$, and $h_{(i+m)(j+n)} \in \mathbb{R}^{B \times C^{\text{in}}}$ represents the units of the input feature map h , and $\omega_{mn} \in \mathbb{R}^{C^{\text{in}} \times C^{\text{out}}}$ represents the convolution weights.

Our EKS Convolution improves upon the traditional convolution operation by enabling gradient separation to be achieved in a single forward propagation, regardless of the number of tasks. EKS Convolution avoids the computational overhead of duplicating data input for each convolution, greatly enhancing training efficiency. Specifically, for each EKS Convolution, in addition to the input feature map h , the task label $\mathbf{M} \in \mathbb{R}^{B \times T}$, corresponding to the mini-batch is also simultaneously inputted as a reference for subsequent parameter aggregation, and \mathbf{M} is a one-hot vector. Then, the output features can be computed by

$$g = g_1 \cup \dots \cup g_t \cup \dots \cup g_T$$

$$g_t = (\mathbf{W}_0 + \mathbf{B}_t \mathbf{A}_t) h_t = (\mathbf{W}_0 + \mathbf{B}_t \mathbf{A}_t) \mathbf{M}_t h, \quad (1)$$

where \cup means the concatenation operation, h_t represents the set of B^t features in h that correspond to the t -th task, and \mathbf{M}_t is an index matrix that indicates which features in h belong to the t -th task. To avoid redundant convolutional operations, we propose parameter aggregation, where the parameters used in the current iteration are aggregated into \mathbf{W}' according to \mathbf{M} . This ensures that the number of forward propagation is always equal to 1, and the operation $g = \mathbf{W}'h$ is equivalent to the Eqn. (1). Specifically, the operation of the Eqn. (1) can be converted by

$$\begin{aligned} g &= (\mathbf{W}_0 + \mathbf{B}_1\mathbf{A}_1)h_1 \cup \dots \cup (\mathbf{W}_0 + \mathbf{B}_T\mathbf{A}_T)h_T \\ &= (\mathbf{W}_0 + \sum_{i=1}^T (\widetilde{\mathbf{B}\mathbf{A}} \odot \mathbf{M})_i)h = \mathbf{W}'h, \end{aligned}$$

where $\widetilde{\mathbf{B}\mathbf{A}} \in \mathbb{R}^{1 \times T \times C^{\text{out}} \times C^{\text{in}} \times k \times k}$ contains the weight of all low-rank expert modules, which can be obtained by

$$\widetilde{\mathbf{B}\mathbf{A}} = \mathbf{B}_1\mathbf{A}_1 \cup \dots \cup \mathbf{B}_t\mathbf{A}_t \cup \dots \cup \mathbf{B}_T\mathbf{A}_T.$$

\odot represents the Hadamard product, and $\widetilde{\mathbf{B}\mathbf{A}} \odot \mathbf{M} \in \mathbb{R}^{B \times T \times C^{\text{out}} \times C^{\text{in}} \times k \times k}$ represents the configuration of low-rank expert modules for the each input feature and i corresponds to the second dimension of $(\widetilde{\mathbf{B}\mathbf{A}} \odot \mathbf{M})$. The weight of shared convolution \mathbf{W}_0 is applied to all tasks. In this way, we obtain the aggregated weight $\mathbf{W}' \in \mathbb{R}^{B \times C^{\text{out}} \times C^{\text{in}} \times k \times k}$, which is equivalent to Eqn. (1) but requires only a single forward propagation.

Another challenge associated with it is that \mathbf{W}' has 5 dimensions, unlike traditional convolutions which typically have 4 dimensions. To ensure compatibility with existing deep learning libraries, we have borrowed the concept of group convolution (GConv) [23]. Specifically, we set the group number to B and $\gamma \in \{1, \dots, B\}$. Then, we reshape h to $h \in \mathbb{R}^{1 \times BC^{\text{in}} \times H \times W}$ and reshape \mathbf{W}' to $\mathbf{W}' \in \mathbb{R}^{BC^{\text{out}} \times C^{\text{in}} \times k \times k}$. Consequently, each output feature unit o_{ij} in g can be computed by

$$\begin{aligned} o_{ij} &= o_{ij}^1 \cup \dots \cup o_{ij}^\gamma \cup \dots \cup o_{ij}^B \\ o_{ij}^\gamma &= \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} h_{(i+m)(j+n)}^\gamma \cdot \omega_{mn}^\gamma, \end{aligned} \quad (2)$$

where $h_{(i+m)(j+n)}^\gamma$ and ω_{mn}^γ represent the reshaped version. Eqn. (2) is a standard form of group convolution, which can be easily implemented in existing deep learning libraries. A notation table can be found in the supplementary materials.

3.3. Loss Function

In order to transfer the knowledge from the medical foundation model into the lightweight decomposition model, we introduce a task knowledge transfer loss denoted as $\mathcal{L}_{\text{transfer}}$. Specifically, for a mini-batch of training data $\{(x_i, y_i, y_i^t)\}_{i=1}^B$, where x_i represents the i -th input image in the current mini-batch, y_i represents the class label across all tasks, and y_i^t represents the class label within its cor-

responding task t . We denote the feature extracted from the foundation model as $f_i^b = F(x_i; \theta_{F_p})$, and the features extracted from the lightweight decomposition model as $f_i^d = F(x_i; \theta_{F_s}; \theta_{E_t})$. Then, the $\mathcal{L}_{\text{transfer}}$ for sample x_i can be written as $\mathcal{L}_{\text{KL}}(f_i^b, f_i^d)$, where \mathcal{L}_{KL} represents the Kullback-Leibler divergence.

Moreover, we can also leverage class label information $\{y_i^t\}$ to enhance task-level supervision. Specifically, during training, we integrate T classification heads $\{h_1, \dots, h_T\}$ into the lightweight decomposition model (classifier in Figure 2). These classification heads can individually predict $\{Y_1, \dots, Y_T\}$ classes where Y_t represents the number of classes for the t -th task, Y is the total number of all classes and $\sum_{i=1}^T Y_i = Y$. The logits extracted from the lightweight decomposition model can be denoted as $g_i^d = h_t(f_i^d)$ and the prediction probability can be calculated by

$$p_{i,j}^d = \frac{\exp(g_{ij}^d/\alpha)}{\sum_{j=1}^{Y_t} \exp(g_{ij}^d/\alpha)},$$

where g_{ij}^d represents the j -th logit in g_i^d and α is the temperature. $\mathcal{L}_{\text{CE}}(y_i^t, p_i^d)$ represents the task-level supervision loss of x_i . Then, the total loss can be written as:

$$\mathcal{L}_{\text{total}} = \frac{1}{B} \sum_{t=1}^T \sum_{i=1}^{B^t} [\mathcal{L}_{\text{CE}}(y_i^t, p_i^d) + \beta \alpha^2 \mathcal{L}_{\text{KL}}(f_i^b, f_i^d)],$$

where β is a hyperparameter.

3.4. Task Knowledge Switch

After decomposition, the lightweight decomposition model enables easy switching between different task knowledge through task knowledge switch, allowing for conversion to the corresponding expert model based on the requirements of various medical departments. Specifically, when deploying the model on the t -th task, the original parameters \mathbf{W}_0 can be replaced with $\mathbf{W}_t = \mathbf{W}_0 + \mathbf{B}_t\mathbf{A}_t$. Similarly, when switching knowledge to another task t' , expert knowledge can be conveniently switched using $\mathbf{W}_0 = \mathbf{W}_t - \mathbf{B}_t\mathbf{A}_t$ and $\mathbf{W}_{t'} = \mathbf{W}_0 + \mathbf{B}_{t'}\mathbf{A}_{t'}$. The parameter fusion mechanism of low-rank expert modules ensures that the deployed expert models consistently maintain a size equal to F_s .

4. Experiments

4.1. Experimental Setup

Dataset. To evaluate the decomposition performance, we use three medical multi-task datasets with different data scales: Radimagenet [30] (1.35 million images), MedMnist [51] (705,689 images) and Med-MT (119,655 images). We decompose the foundation models pre-trained on these datasets into 11/10/8 lightweight expert models, respectively, with each decomposed expert model focusing on a specific anatomical region. Detailed information about these datasets can be found in the supplementary materials.

In addition, to determine the extent to which the decomposed expert models can fully replace pre-trained models in specific domains, we evaluated the transferability of these expert models on seven different downstream datasets, including COVID [50], BTC [37], AD [1], Mura [34], AITD [2], HAM10000 [45], and DET10 [24]. Detailed information can be found in the supplementary materials.

Competitive methods. (1) **Baseline** refers to training from scratch on downstream tasks. (2) **Single-Task Learning (STL)** refers to training multiple single-task networks independently. (3) **Multi-Task Learning (MTL)** refers to training a single model to predict all tasks. (4) **STL-KD** and (5) **MTL-KD** correspond to the KD version of STL and MTL, respectively, which utilize knowledge distillation to transfer knowledge from the pre-trained models. (6) **MoCo-MTL** [13] and (7) **Aligned-MTL** [41] are the advanced MTL algorithms. (8) **KF** [52] represents the advanced knowledge decomposition method, which is the closest to our goal and serves as our primary comparison object. We explain the purpose of using these methods for comparison in the supplementary materials.

Implementation details. For the decomposition training, we use the SGD optimizer with a learning rate of 0.05 and CosineAnnealingLR as the scheduler for training 100 epochs. For the downstream fine-tuning, we use AdamW optimizer with a learning rate of 5e-5 and train the model for 240 epochs. The default values for the hyperparameters are set as follows: $\alpha=10$, $\beta=1$ and $r=8$. The pre-trained model structure is ResNet50 [16], and the structure of the lightweight decomposition model is ShuffleNetV2 [28]. In our experiments, we apply EKS convolution to all convolutions in ShuffleNetV2 except for GConv.

4.2. Decomposition performance

The performance comparison of different methods on three pre-training datasets is shown in Table 1. Each column corresponds to a specific task. ‘‘Avg’’ represents the task-level average accuracy. ‘‘Parmas’’ represents the total number of parameters during training. Only KF and our method focus on the knowledge decomposition of pre-trained models. Considering the generalization requirement of the foundation model, it is common for the foundation model to use a unified classification head during training, instead of configuring a specific classification head for each task [30]. This is also why the performance of the foundation model in Table 1 is relatively poor. Note that other methods can only solve problems within specific tasks.

The foundation model vs. STL. The results of the foundation model are superior to STL on Med-MT, but significantly inferior to STL on Radimagenet and MedMnist, particularly for MedMnist, which suggest that as the increase of scale and diversity of the pre-training dataset, the specialization of the pre-trained model gradually diminishes due to

conflicts between different domain knowledge. In contrast, training models independently for each task (STL) can prevent interference between different tasks and thus achieve better performance than foundation models on Radimagenet and MedMnist. However, STL is unable to learn common knowledge among tasks, so it often requires more data to ensure its generalization ability. Besides, training T individual models is not only time-consuming but also results in a linear increase of parameters.

MTL-based methods vs. STL-based methods. We can also observe that MTL outperforms STL in Radimagenet and Med-MT, while underperforms STL in MedMnist. It may be related to the degree of correlation between tasks included in the pre-training dataset, where MedMnist has the most diverse modalities (refer to supplementary materials). In contrast to the standard MTL, other advanced MTL methods, namely MoCo-MTL and Aligned-MTL, do not yield improvements and may even exhibit worse performance. This observation suggests that balancing multiple optimization objectives to obtain a better shared encoder is not an effective solution when there are significant differences among tasks. The knowledge distillation variants of STL and MTL (STL-KD and MTL-KD) do not show significant performance improvement, which suggests that the general features extracted by foundation models have limited benefits for specific tasks and indirectly reflects the importance of both specialized and general features. It aligns with the design philosophy of our LoRKD.

LoRKD vs. KF and other methods. Compared to the knowledge decomposition method KF, our method shows significant performance advantages and introduces fewer parameters. Specifically, even including 11/10/8 experts, our method has less than half the number of parameters of KF. This result validates the effectiveness of our low-rank expert modules and the efficient knowledge separation convolution. Furthermore, our method also achieves the best average performance compared to other non-knowledge decomposition baselines, highlighting the potential of knowledge decomposition in extracting task-specific knowledge.

4.3. Transferability

In order for the decomposed lightweight expert model to fully replace the foundation model within a specific domain, it is necessary not only for the expert models to perform well on the same distribution of data (pre-training dataset), but also to evaluate its generalization ability on downstream tasks with close distributions. The performance comparison of the expert models decomposed from three pre-training datasets on seven downstream datasets is shown in Table 2.

For KF and our method, we fine-tune the corresponding expert models on downstream datasets, such as using the lung expert model for the COVID dataset. If there was no corresponding expert model, similar to [52], we fine-tune on

Table 1. The decomposition performance on pre-training datasets. Each column represents the performance of different methods for specific tasks. It is worth noting that except for KF and ours, the concept of knowledge decomposition does not exist in other methods. The presence of homonymous experts implies different modalities. For more details, please refer to the supplementary materials.

Radimagenet (1.35 million images, 11 tasks)													
Method	Params(M)	Lung	Abdomen	Thyroid	Abdomen	Knee	Shoulder	Spine	Ankle	Abdomen	Brain	Hip	Avg
Foundation	23.51	36.22	46.52	74.05	48.42	40.09	31.32	17.79	12.95	64.17	77.30	32.33	43.74
STL	13.79	76.42	33.94	91.55	69.17	49.32	41.80	20.62	20.31	65.99	83.88	51.05	54.91
MTL	1.25	77.16	37.45	91.73	68.43	46.47	42.72	20.85	18.17	71.13	84.67	55.16	55.81
STL-KD	13.79	78.00	31.74	91.34	69.10	46.57	43.09	19.77	19.43	69.85	83.83	52.19	54.99
MTL-KD	1.25	78.92	33.89	91.97	68.54	48.51	43.34	21.03	18.48	69.58	84.18	54.90	55.75
MoCo-MTL	1.25	76.28	45.56	86.26	67.00	45.58	43.97	18.74	17.41	74.88	84.33	52.71	55.70
Aligned-MTL	1.25	77.74	36.38	91.76	68.51	48.41	43.28	21.26	18.37	68.57	84.54	54.86	55.79
KF	5.01	64.57	20.38	95.82	68.05	45.56	39.03	24.18	16.69	56.65	78.46	51.74	51.01
LoRKD	2.21	78.72	36.95	91.87	68.77	48.80	43.26	21.41	19.26	69.24	84.60	55.93	56.26

MedMnist (705,689 images, 10 tasks)												
Method	Params(M)	Colon	Retinal	OrganC	Cell	Breast	Tissue	Skin	OrganA	OrganS	Chest	Avg
Foundation	23.51	87.41	77.40	23.51	50.37	84.62	40.55	12.92	18.64	18.90	86.22	50.05
STL	12.54	84.53	78.40	89.65	96.81	85.26	68.89	73.97	92.90	77.43	85.42	83.33
MTL	1.25	80.99	77.10	89.90	95.67	83.33	65.42	74.21	91.33	76.34	86.89	82.12
STL-KD	12.54	84.33	77.10	90.45	96.52	83.33	68.25	74.81	93.53	77.52	82.53	82.84
MTL-KD	1.25	82.83	75.20	90.02	95.94	83.26	64.56	74.31	92.13	76.02	86.39	82.06
MoCo-MTL	1.25	76.10	69.80	80.00	86.55	76.92	63.89	69.18	83.82	67.81	83.87	75.79
Aligned-MTL	1.25	79.78	73.10	89.70	95.44	88.46	64.00	74.36	90.81	75.06	86.22	81.69
KF	4.67	37.83	48.20	72.40	44.93	80.13	54.17	38.01	71.75	59.19	72.12	57.87
LoRKD	2.12	83.90	78.60	90.57	96.26	87.18	67.01	73.97	92.83	77.27	87.39	83.50

Med-MT (119,655 images, 8 tasks)											
Method	Params(M)	Retinal	Skin	Breast	GI tract	Lung	Shoulder	Lung	Bone	Avg	
Foundation	23.51	81.83	87.01	81.82	91.25	66.37	92.31	65.00	59.46	78.13	
STL	10.03	75.27	77.92	76.59	85.62	69.91	75.00	64.85	51.15	72.04	
MTL	1.25	78.14	78.57	77.85	87.94	69.91	79.81	64.37	49.41	73.25	
STL-KD	10.03	71.45	67.53	77.18	86.06	60.18	78.85	64.67	51.23	69.64	
MTL-KD	1.25	79.23	77.27	77.89	88.06	76.11	77.88	64.84	49.17	73.80	
MoCo-MTL	1.25	58.74	55.84	51.74	48.31	67.26	67.31	46.76	20.11	52.01	
Aligned-MTL	1.25	61.07	56.49	51.50	52.63	69.03	67.31	46.77	19.17	53.00	
KF	3.99	65.30	74.67	52.19	61.12	77.88	79.81	60.21	33.50	63.09	
LoRKD	1.95	79.37	85.06	79.04	88.63	72.57	83.65	65.07	52.42	75.73	

the shared backbone (with \dagger). As for other non-knowledge decomposition methods, we use the models trained on the pre-training dataset for fine-tuning to demonstrate the advantages of knowledge decomposition in terms of transferability. For details, please refer to supplementary materials.

The performance of fine-tuning foundation models is observed to be inferior to Baseline, providing evidence again that the foundation model cannot replace task-specific models in terms of performance, due to lack of specialization. Compared to baseline, both STL-based and MTL-based methods show little improvement, indicating that solely focusing on task-specific knowledge or common knowledge does not contribute to transferability. Conversely, our expert models incorporate both common knowledge and task-specific knowledge, which exhibit strong transferability and even significantly outperform KF. Another benefit when compared to KF is that our method is compatible with the

parameter fusion and does not require the simultaneous deployment of two networks (CKN and the corresponding TSN need to be deployed simultaneously in KF).

Furthermore, we discover an interesting phenomenon. In comparison to MTL-KD, our method outperforms it more significantly on downstream datasets. This shows the advantage of knowledge decomposition in transferability, and the transferability can not be directly reflected through the decomposition performance. And as the scale of the pre-training dataset increases, the transferability of our decomposed expert models also improves, indicating that increasing the scale of pre-training datasets benefits the transferability of the decomposed model.

4.4. Ablation Studies and More Analysis

The impact of Rank r . We evaluate the impact of r for decomposition performance on pre-training dataset and trans-

Table 2. The performance of the decomposed expert models on seven downstream datasets. ‘‘Params’’ represents the number of model parameters during deployment. ‘‘Comp. Ratio’’ denotes the compression ratio, defined as the ratio of the deployed model parameters to the parameters of the foundation model. ‘‘-’’ indicates the absence of data corresponding to the downstream tasks in the pre-training dataset.

Pre-train	Model	Params(M)	Comp. Ratio	COVID	BTC	AD	Mura_s	AUITD	HAM10000	DET10	Avg
Radimagenet	Foundation	23.51	/	78.33	80.20	74.35	71.05	96.66	75.08	86.69	80.34
	Baseline	1.25	5.32%	82.76	75.38	76.08	76.73	97.77	74.42	87.54	81.52
	STL	1.25	5.32%	82.76	78.93	76.70	77.26	97.77	-	87.52	-
	MTL	1.25	5.32%	83.25	79.95	74.67	76.91	97.77	75.83	86.82	82.17
	STL-KD	1.25	5.32%	82.27	80.46	76.31	76.73	96.66	-	87.25	-
	MTL-KD	1.25	5.32%	81.77	78.93	73.89	76.55	96.66	74.37	87.17	81.33
	MoCo-MTL	1.25	5.32%	78.82	78.68	69.27	75.49	91.64	71.77	86.43	78.87
	Aligned-MTL	1.25	5.32%	82.27	78.43	70.29	76.91	88.58	73.07	86.91	79.49
	KF	1.60	6.81%	80.79	79.70	71.23	74.96	96.66	74.12 [†]	87.17	80.66
	LoRKD	1.25	5.32%	86.21	81.47	79.12	79.57	98.33	76.03[†]	88.50	84.18
MedMnist	Foundation	23.51	/	80.30	77.41	72.09	76.38	88.86	72.12	86.80	79.14
	Baseline	1.25	5.32%	82.76	75.38	76.08	76.73	97.77	74.42	87.54	81.52
	STL	1.25	5.32%	83.25	-	-	-	97.77	71.82	87.56	-
	MTL	1.25	5.32%	81.28	78.68	77.17	76.19	97.77	74.82	87.36	81.90
	STL-KD	1.25	5.32%	79.80	-	-	-	97.49	73.87	86.93	-
	MTL-KD	1.25	5.32%	80.79	78.62	76.62	75.84	98.05	73.87	87.23	81.57
	MoCo-MTL	1.25	5.32%	78.82	77.16	77.80	74.95	97.77	72.77	86.82	80.87
	Aligned-MTL	1.25	5.32%	82.27	77.42	77.72	76.90	96.37	73.87	86.97	81.65
	KF	1.60	6.81%	80.79	77.15 [†]	72.71 [†]	74.77 [†]	96.10	72.97	87.41	80.27
	LoRKD	1.25	5.32%	84.24	79.70[†]	79.05[†]	77.80[†]	98.33	74.82	87.60	83.08
Med-MT	Foundation	23.51	/	82.76	78.17	69.19	71.76	89.69	75.53	86.69	79.11
	Baseline	1.25	5.32%	82.76	75.38	76.08	76.73	97.77	74.42	87.54	81.52
	STL	1.25	5.32%	-	-	-	-	-	73.77	-	-
	MTL	1.25	5.32%	82.76	76.65	77.48	77.09	97.49	74.92	87.15	81.93
	STL-KD	1.25	5.32%	-	-	-	-	-	74.42	-	-
	MTL-KD	1.25	5.32%	82.76	75.89	74.43	76.91	97.77	74.32	87.34	81.34
	MoCo-MTL	1.25	5.32%	80.79	76.40	77.48	76.91	97.49	72.62	86.91	81.23
	Aligned-MTL	1.25	5.32%	79.80	75.63	76.62	76.73	97.77	73.72	87.19	81.06
	KF	1.60	6.81%	80.79 [†]	74.87 [†]	75.76 [†]	76.73 [†]	98.05 [†]	73.92	87.39 [†]	81.07
	LoRKD	1.25	5.32%	83.25[†]	77.66[†]	76.94 [†]	78.33[†]	98.33[†]	75.18	87.84[†]	82.50

Table 3. The impact of r on performance and transferability.

	Pre-train	COVID	BTC	AD	Mura_s	AUITD	Avg
$r=4$	55.08	85.71	79.95	75.61	77.98	98.05	83.46
$r=8$	56.26	85.71	81.47	75.92	78.51	98.33	84.93
$r=16$	56.19	86.21	82.49	78.81	78.51	98.33	84.87

ferability on downstream datasets in Table 3. The results show that for Radimagenet, increasing r from 4 to 8 leads to a significant performance improvement, while increasing r from 8 to 16 does not provide further enhancement. This suggests that selecting a larger r is not necessarily better and may be related to the scale of datasets. Moreover, we find that the improvement on pre-training dataset is positively correlated with the improvement in transferability.

Lower Costs and Higher Efficiency. We compare the number of parameters and the FLOPs among different methods on Radimagenet, as shown in Table 4. It can be observed that, compared to the foundation model, ours significantly reduce the number of parameters and FLOPs, demonstrating that our method can effectively reduce de-

Table 4. The comparison of the costs of different methods.

Costs	Baseline	KF	LoRKD			Foundation
			$r=4$	$r=8$	$r=16$	
Params (Training)	1.25	5.01	1.73	2.21	3.16	23.51
FLOPs (Training)	0.15	0.63	0.21	0.27	0.38	4.13
Params (Deployment)	1.25	1.60	1.25	1.25	1.25	23.51
FLOPs (Deployment)	0.15	0.16	0.15	0.15	0.15	4.13

ployment costs while maintaining high computational efficiency. In comparison to KF, even at $r=16$, our method still incurs significantly lower costs. As r increases, our costs do not increase significantly and remains at a lower level. It is worth mentioning that if parameter fusion is used, our costs will be the same as baseline. Furthermore, we compare the efficiency of different methods in Table 5 following the setting in [48], where b , r , l and d represent the batch size, the rank, the maximum sequence length and the feature dimension respectively. c_2 is the computational coefficient of matrix multiplication. If the following condition is satisfied, $\frac{rbd^2}{Td^2r+bd^2l} \geq 1 \implies \frac{Tr}{bl} + 1 \leq r$, the efficiency of EKS conv is higher. This inequality holds true in most real-world

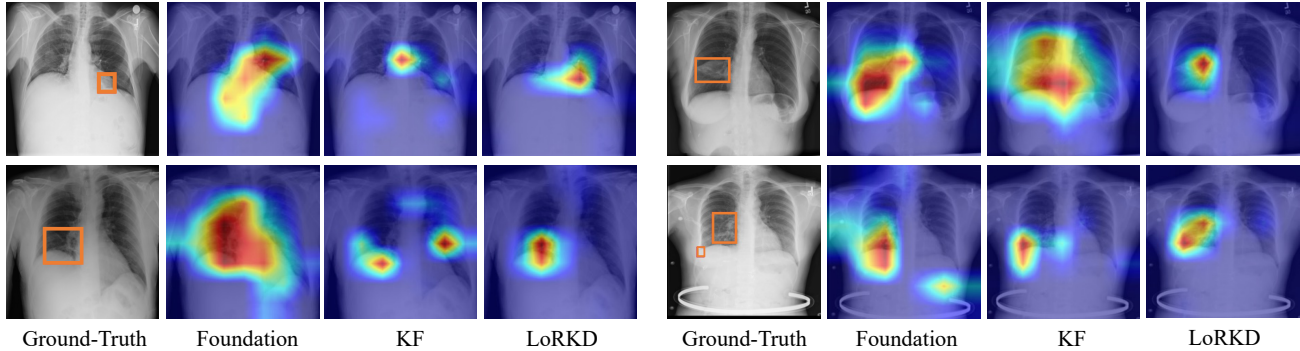


Figure 3. Comparison of Grad-CAM visualizations between the decomposed model and the foundation model on DET10. The foundation model tends to focus on larger regions, corresponding to its general feature extraction capability, while our decomposed expert model focuses on more precise regions, reflecting stronger specialization.

Table 5. Efficiency comparison of different methods for constructing personalized low-rank experts for each sample.

Method	Improved Operation	Computational Cost
FLoRA [48]	$\mathbf{Y} = \mathbf{A} \circ ((\mathbf{B} \circ \mathbf{X})\mathbf{W}_0)$	$c_2(rbl d^2)$
EKS conv (ours)	$\mathbf{Y} = \mathbf{X}(\mathbf{W}_0 + \sum_{i=1}^T (\widehat{\mathbf{B}}\mathbf{A} \odot \mathbf{M}_i))$	$Tc_2(rd^2) + c_2(bld^2)$

cases, as $bl > Tr$, $r > 2$ are common settings. A detailed explanation is provided in the supplementary materials.

Stronger Specialization. Taking the DET10 dataset with detection annotations as an example, we evaluate the differences in the activated regions between the decomposed expert model and the foundation model during the prediction process from the perspective of Grad-CAM [39]. The corresponding visualization results are shown in Figure 3.

It can be observed that although the foundation model can focus on the correct regions, the range of regions it attends to is usually larger compared to the detection boxes of Ground-Truth. This may be due to the fact that the feature extractor of the foundation model retains a certain degree of general feature extraction ability, tending to focus on more information regardless of the specific task. Conversely, our expert models focus on more precise abnormal regions and demonstrate stronger specialization compared to the foundation model. Compared to KF, our approach also achieves higher recognition accuracy, further indicating that our decomposed model has better transferability.

Knowledge Disentanglement. To verify whether our method can indeed achieve knowledge decoupling between different tasks, we measure the mutual information gap (MIG) scores [8] for different methods. MIG is a widely used metric for quantifying disentanglement. The results are shown in Figure 4, where higher MIG scores indicate a higher level of disentanglement. It can be observed that our method exhibits a higher level of disentanglement compared to the previous KF and other baselines, which may benefit from our explicit gradient separation.

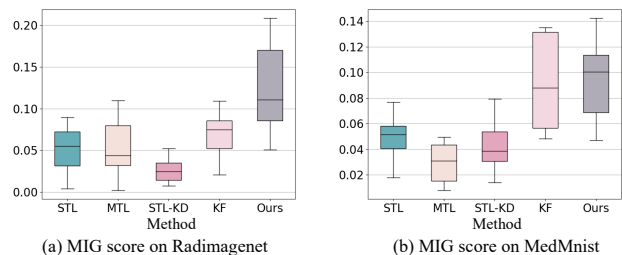


Figure 4. The comparison of MIG scores on different methods.

In addition, we find that the degree of disentanglement in MTL is lower compared to STL. This suggests that when MTL employs a shared encoder to acquire common knowledge for across tasks, the entanglement of gradients from different tasks also results in knowledge entanglement. Additionally, compared to STL, the degree of disentanglement in STL-KD is also lower, which can be attributed to the transfer of common knowledge from the foundation model.

More analysis, such as using larger foundation models and comparing CKA feature similarity across different tasks, can be found in the supplementary materials.

5. Conclusion

This paper proposes a new perspective called knowledge decomposition, which focuses on reducing the deployment costs and enhancing specialization for medical foundation models. We design a low-rank expert module and an efficient gradient separation convolution to successfully decompose the foundation model into multiple lightweight expert models, and validate their transferability and disentanglement. We hope that this research will provide new insights for the development of medical foundation models.

Acknowledgment. This work is supported by the National Key R&D Program of China (No. 2022ZD0160703), STCSM (No. 22511106101, No. 22511105700, No. 21DZ1100100), 111 plan (No. BP0719010) and National Natural Science Foundation of China (No. 62306178).

References

- [1] Alzheimer’s dataset, Kaggle dataset. <https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images>. 5
- [2] Algerian ultrasound images thyroid dataset: Auitd,Kaggle dataset. <https://www.kaggle.com/datasets/azouzmaraoua/algeria-ultrasound-images-thyroid-dataset-aitd>. 5
- [3] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9163–9171, 2019. 2
- [4] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1
- [5] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in backslash beta-vae. *arXiv preprint arXiv:1804.03599*, 2018. 2
- [6] R Caruana. Multitask learning: A knowledge-based source of inductive bias1. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Citeseer, 1993. 2
- [7] Rich Caruana. Multitask learning. *Machine learning*, 28: 41–75, 1997. 2
- [8] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018. 8
- [9] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016. 2
- [10] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020. 2
- [11] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012. 2
- [12] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. *arXiv preprint arXiv:2101.04731*, 2021. 2
- [13] Heshan Devaka Fernando, Han Shen, Miao Liu, Subhajit Chaudhury, Keerthiram Murugesan, and Tianyi Chen. Mitigating gradient bias in multi-objective learning: A provably convergent approach. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 5
- [14] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018. 2
- [15] Ben Glocker, Charles Jones, Mélanie Roschewitz, and Stefan Winzeck. Risk of bias in chest radiography deep learning foundation models. *Radiology: Artificial Intelligence*, 5(6): e230060, 2023. 1
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5
- [17] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2016. 2
- [18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2
- [19] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 2
- [20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2
- [21] Yuhao Huang, Xin Yang, Lian Liu, Han Zhou, Ao Chang, Xinrui Zhou, Rusi Chen, Junxuan Yu, Jiongquan Chen, Chaoyu Chen, et al. Segment anything model for medical images? *arXiv preprint arXiv:2304.14660*, 2023. 1
- [22] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 2
- [23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 4
- [24] Jingyu Liu, Jie Lian, and Yizhou Yu. Chestx-det10: Chest x-ray dataset on detection of thoracic abnormalities, 2020. 5
- [25] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019. 2
- [26] Yang Liu, Zhaowen Wang, Hailin Jin, and Ian Waisell. Multi-task adversarial network for disentangled feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3743–3751, 2018. 2
- [27] Sindy Löwe, Peter O’Connor, and Bastiaan S Veeling. Greedy infomax for self-supervised representation learning. 2019. 2
- [28] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 5
- [29] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adver-

- sarial training. *Advances in neural information processing systems*, 29, 2016. 2
- [30] Xueyan Mei, Zelong Liu, Philip M Robson, Brett Marinelli, Mingqian Huang, Amish Doshi, Adam Jacobi, Chendi Cao, Katherine E Link, Thomas Yang, et al. Radimagenet: an open radiologic deep learning research dataset for effective transfer learning. *Radiology: Artificial Intelligence*, 4(5): e210315, 2022. 4, 5
- [31] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, pages 5191–5198, 2020. 2
- [32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2
- [33] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3967–3976, 2019. 2
- [34] Pranav Rajpurkar, Jeremy Irvin, Aarti Bagul, Daisy Ding, Tony Duan, Hershel Mehta, Brandon Yang, Kaylie Zhu, Dillon Laird, Robyn L Ball, et al. Mura: Large dataset for abnormality detection in musculoskeletal radiographs. *arXiv preprint arXiv:1712.06957*, 2017. 5
- [35] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 2
- [36] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 2
- [37] Ahmad Saleh, Rozana Sukaik, and Samy S. Abu-Naser. Brain tumor classification using deep learning. In *2020 International Conference on Assistive and Rehabilitation Technologies (iCareTech)*, pages 131–136, 2020. 5
- [38] Andrew B Sellaergren, Christina Chen, Zaid Nabulsi, Yuanzhen Li, Aaron Maschinot, Aaron Sarna, Jenny Huang, Charles Lau, Sreenivasa Raju Kalidindi, Mozziyar Etemadi, et al. Simplified transfer learning for chest radiography models using less data. *Radiology*, 305(2):454–465, 2022. 1
- [39] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 8
- [40] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018. 2
- [41] Dmitry Senushkin, Nikolay Patakin, Arseny Kuznetsov, and Anton Konushin. Independent component alignment for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20083–20093, 2023. 2, 5
- [42] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019. 2
- [43] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 2
- [44] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424, 2017. 2
- [45] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018. 5
- [46] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1365–1374, 2019. 2
- [47] Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobzyev, and Ali Ghodsi. Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *arXiv preprint arXiv:2210.07558*, 2022. 2
- [48] Yeming Wen and Swarat Chaudhuri. Batched low-rank adaptation of foundation models. *arXiv preprint arXiv:2312.05677*, 2023. 7, 8
- [49] Chaoyi Wu, Jiayu Lei, Qiaoyu Zheng, Weike Zhao, Weixiong Lin, Xiaoman Zhang, Xiao Zhou, Ziheng Zhao, Ya Zhang, Yanfeng Wang, et al. Can gpt-4v (ision) serve medical applications? case studies on gpt-4v for multimodal medical diagnosis. *arXiv preprint arXiv:2310.09909*, 2023. 1
- [50] Yang Xingyi, He Xuehai, Zhao Jinyu, Zhang Yichen, Zhang Shanghang, and Xie Pengtao. Covid-ct-dataset: a ct image dataset about covid-19. *arXiv preprint arXiv:2003.13865*, 2020. 5
- [51] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023. 4
- [52] Xingyi Yang, Jingwen Ye, and Xinchao Wang. Factorizing knowledge in neural networks. In *European Conference on Computer Vision*, pages 73–91. Springer, 2022. 1, 2, 5
- [53] Zhendong Yang, Ailing Zeng, Zhe Li, Tianke Zhang, Chun Yuan, and Yu Li. From knowledge distillation to self-knowledge distillation: A unified approach with normalized loss and customized soft labels. *arXiv preprint arXiv:2303.13005*, 2023. 2
- [54] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 2
- [55] Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*, 2023. 2
- [56] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021. 2