On the Inductive Bias Transfer with Knowledge Distillation

Byeong Tak Lee
MedicalAI, Inc.
Seoul, South Korea
bytaklee@medicalai.com

Joon-myoung Kwon
MedicalAI, Inc.
Seoul, South Korea
cto@medicalai.com

Yong-Yeon Jo
MedicalAI, Inc.
Seoul, South Korea
yy.jo@medicalai.com

ABSTRACT

In the lack of data, an appropriate inductive bias is ones of key factors for the successful training of a model. One approach to transfer inductive bias between different structure of networks is to utilize knowledge distillation. Several studies achieve promising result in the several computer vision datasets using response-based knowledge distillation. However, we observe that the previous method fails to transfer inductive bias when the dataset contains fewer datapoints or classes. To solver the problem, we propose to use feature-based knowledge distillation for effective inductive bias transfer. Through extensive experimentation and analysis, we demonstrate that the suggested method is capable of transferring inductive bias and outperforming previous methods.

KEYWORDS

Inductive bias, Knowledge distillation, Electrocardiogram, Electronic health record

1 INTRODUCTION

Inductive biases are constraints enforcing the model to have specific properties [1, 5]. The effect of an appropriate inductive bias is comparable to the effect of additional data; in other words, one can compensate for the lack of data by exploiting strong inductive biases [1, 5]. Nevertheless, such constraints are not always advantageous. If the inductive bias induced by architecture of the network is too restrictive, the model can only learn limited representations [5].

Transferring the inductive bias is realized through the knowledge distillation [1] which make a student network to encode properties of a teacher network. For example, Data-efficient image Transformers (DeiT) uses the convolutional neeural network to inherit its inductive bias to transformer network. It uses the distillation token to predict the output of the pre-trained convolutional neural network, achieving the performance on par with the model already trained with a strong inductive bias [13].

We wonder the applicability of transferring the inductive bias via the knowledge distillation in the various real-world datasets. We evaluated the technique transferring the inductive bias in used DeiT on two types of medical datasets: the inductive biases (1) in convolutional neural networks (CNN) on the electrocardiograms (ECG) and in (2) recurrent neural networks (RNN) on the electronic health records (EHR). As shown in Figure 1, we observe that the performance of the transformer trained with DeiT is significantly inferior to that of the teacher networks.

Our contributions in this study are the following: First, We analyze the limitation of the previous methods of transferring the inductive bias through the knowledge distillation. Second, We examine the reason for the failure of the previous methods via rigorous experiments. Third, Based on the findings from the experimental results, we propose an effective way to transfer inductive biases through the knowledge distillation.

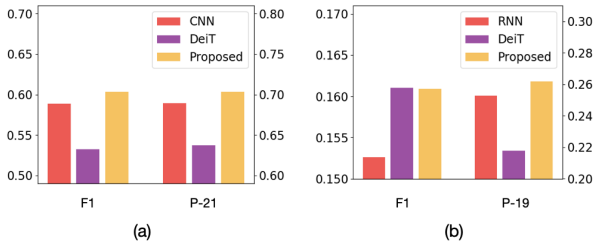

Figure 1: Performance in (a) ECG from Physionet 2021 and (b) EHR from Physionet 2019. F1 refers to the f-1 score (the left side of the *y*-axis), and P-19/P-21 indicate the physionet 19 and physionet 21 scores (the right side of the *y*-axis).

2 DEMYSTIFYING INDUCTIVE BIAS ENCODED IN THE STUDENT NETWORK

If a transformer encodes the inductive biases of its teacher network, then the its representations or filters exhibit characteristics of the inductive bias. In this perspective, we investigate the representations and the self-attention layers of the pre-trained transformer.

2.1 Experiment setting

2.1.1 Dataset. We used following datasets with different properties: Physionet 2021 for CNN and Physionet 2019 for RNN. Physionet 2021 is public ECG datasets [12], which contains approximately 88,000 ECGs. Each ECG is assigned one or more arrhythmia labels, for a total of 26 classes of arrhythmia [12]. PhysioNet 2019 [11] is a EHR consisting of hourly clinical variables collected from the intensive care unit (ICU) of two hospital systems with a total number of 40,336 patients The task is to predict sepsis within 12 hours, and the onset of sepsis is given to each patient.

We additionally used two external datasets to see if the DeiT preserves the inductive biases of CNN/RNN regardless of the data distribution. The Hangzhou dataset [2] contains 20,036 ECG recordings, and the eICU Collaborative Research Database [10] is a multicenter database containing over 200,000 admissions to ICU.

2.1.2 Architecture. We develop two teacher networks: (1) the ResNet-based network for ECG datasets [6]. Each block of ResNet contains two layers of convolution, and there are eight blocks in total. The architectural detail is identical to Hannun et al [6]. (2) the long short time memory (LSTM) network for EHR datasets [15]. LSTM is stacked with the 3-layer, and each layer have 256 hidden units with a residual connection between each layer.

Table 1: Probability similarity between the CNN/RNN and the transformers trained using different methods.

	ECG		EHR	
	P21	Hanzhou	P19	eICU
Transformer	0.530	0.174	0.439	0.308
DeiT	0.601	0.206	0.464	0.351
BBM	0.796	0.592	0.776	0.728

As a student network, we adopt a transformer [14]. There are two student networks, each of which has eight blocks for the ECG dataset and three blocks for the EHR dataset. Training a transformer on ECG datasets, we split a signal into patches following Dosovitskiy et al. [4]. Each patch consists of 100ms (20 timestamps) without overlapping, and is used as the input of a transformer. In EHR datasets, a patient have multiple rows, each of which consists of medical record on a time. A single row is used as a token of the input.

2.1.3 The other details of experiments. We set a batch size to 512 for ECG dataset and a batch size to 256 for EHR dataset. We use an Adam optimizer with the weight decay and the cosine warmup scheduler that peaks at ten epochs. In the experiment with ECGs, the rand augment policy [3] is adopted with six data augmentation methods including the gaussian smoothing, time resampling and cut, gaussian noise, baseline wander, time mask, and channel mask. In the case of EHRs, data augmentation is not applied. Hyperparameters, such as the learning rate, weight decay, dropout, and parameters for the augment policy, are randomly selected from predefined search space, tuned by the asynchronous successive halving algorithm [8] using the ray framework [9]. Train, validation, and test set are divided into a ratio of 0.7:0.15:0.15.

2.2 Representation analysis

In order to analyze the inductive bias induced by the structure of networks, we compare the representations of the teacher and the student networks. If the student model successfully encodes the inductive bias of the teacher model, there are similarities in the representations between them.

2.2.1 Output similarity. We first examine the output similarity using the r-square values. As shown in Table 1, the similarity between DeiT and its teacher (CNN/RNN) is slightly higher than the similarity between the naive transformer and CNN/RNN, however, the discrepancy between the teacher and the student is still large. This implies the possibility of failure of DeiT encoding the inductive bias of its teacher architecture.

2.2.2 Internal representation similarity. To examine internal representation similarity driven by the architecture, we exploit central kernel analysis [7]. Figure 2 illustrates the representational similarity between CNN/RNN in comparison to DeiT, and Transformer. We observe that CNN/RNN's feature extraction process differs from that of Transformer. DeiT has higher similarity to CNN/RNN compared to Transformer, but there is still a substantial difference to its teacher. Specifically, in the case of DeiT, only the early layers exhibit a significant dissimilarity between the representations, indicating that the early layers of DeiT failed to learn the CNN/RNN representation.

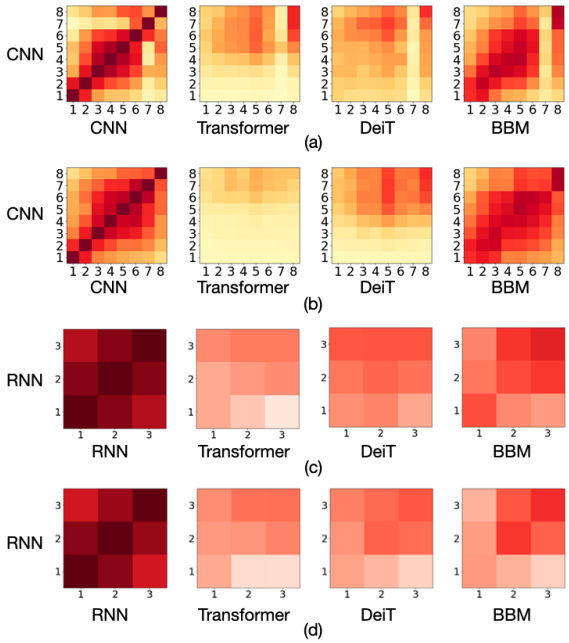

Figure 2: The representational similarity between the CNN/RNN and the transformers. Axes of each matrix represent the order of blocks. (a), (b), (c), and (d) are similarity in Physionet 2021, Hangzhou, Physionet 2019, and eICU dataset.

2.3 Self-attention analysis

If the inductive bias of CNN/RNN is appropriately transferred to the Transformer, the Transformer's self-attention should exhibit the pattern of CNN/RNN, i.e., spatial/temporal invariance and locality. Figure 3 depicts the averaged self-attention matrices in each block across all samples and heads. It is difficult to distinguish the pattern of DeiT distinct from Transformer. To elaborate, DeiT does not exhibit the characteristics that demonstrate the inductive bias of CNN/RNN, such as translational/temporal invariance or locality.

2.4 Discussion

In the experiment, we found no evidence that CNN/RNN's inductive bias is transferred to DeiT. There could be several reasons why DeiT works with ImageNet but not with our dataset. The first possibility is the size of the dataset. In the case of ImageNet, large data of 1M is sufficient to transfer inductive bias via KD. However, the size of data we utilized is only 8 percent of ImageNet, so it may be challenging to transfer inductive bias via KD. The second possibility is the number of classes. ImageNet consists of one thousand classes, whereas the dataset we utilized consists of twenty-six for Physionet 2021 and two classes for Physionet 2019. With this respect, DeiT may not work with our dataset because distributions obtained from our dataset contain less information than distribution obtained from

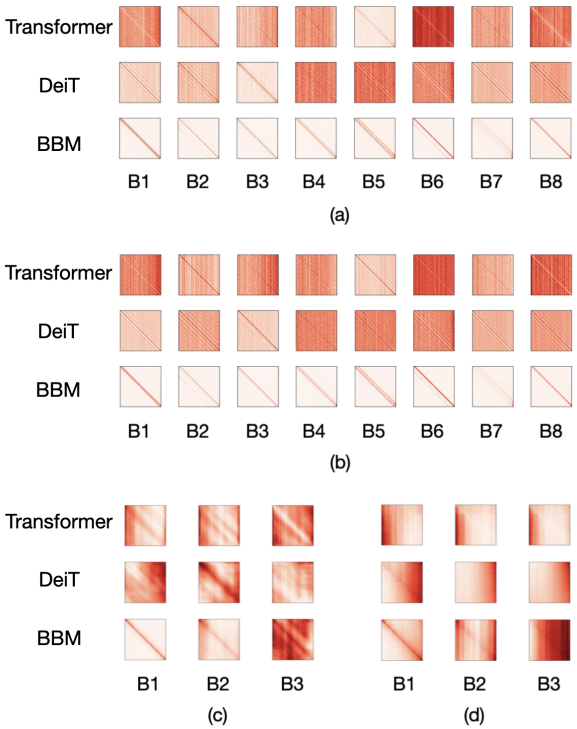

Figure 3: The self-attention of the transformers. Each (a), (b), (c), and (d) is self-attention matrix in Physionet 2021, Hangzhou, Physionet 2019, and eICU dataset. BN indicates the self-attention matrix at N th layer.

ImageNet. Based on this, we believe the problem can be alleviated if the student is provided with more information to encode inductive bias.

3 BETTER SOLUTION FOR TRANSFERRING INDUCTIVE BIAS

3.1 Feature-based knowledge distillation

The knowledge distillation utilized in the DeiT is a type of response-based knowledge distillation, which distill knowledge using output of the model. In contrast to the previous works, we impose a stronger signal by using feature-based knowledge distillation to enforce the student network to learn the teacher's inductive bias. Additionally, knowledge distillation is performed on feature maps in order to effectively transfer spatial information from the teacher to the student.

First, we divide the teacher (g) and student (f) into the same number of blocks, and then perform the knowledge distillation between corresponding blocks (f_n, g_n) of the teacher and the student. Since features transverse multiple layers, the dimension of it varies. For example, in the case of CNN, the pooling operation and convolution with stride change the dimensions with temporal direction, and convolution operation change also increase the dimension of feature. Because of this, the dimension of features used for knowledge distillation can vary. To solve the problem, we introduce a transformation function (h) that transforms each dimension to be identical. This function resizes the feature's dimensions along the temporal axis and projects them along the depth axis.

$$h^{f\to g}(z) = I(z)\, W \tag{1}$$

where $h(\cdot) := \mathbb{R}^{t\times d} \to \mathbb{R}^{t'\times d'}$ consist of two layer: $I(\cdot) := \mathbb{R}^{t\times d} \to \mathbb{R}^{t'\times d}$ represents the resize along the temporal axis, and $W \in \mathbb{R}^{d\times d'}$ is linear transformation along the depth axis.

With a transformation function, we match and train each block of the teacher and the student (f_n, g_n) to be similar as illustrated in Figure 4. In addition to matching between blocks of the teacher and the student, we also perform knowledge distillation between the output of the successive composition of blocks of the teacher and the student ($f_n \circ \cdots \circ f_1, g_n \circ \cdots \circ g_1$). Each loss function term is formulated as follows.

$$\mathcal{L}_n^1 = \sum_{t,d} \left\| f_n(z_{n-1}^f) - (h_n^{g\to f} \circ g_n \circ h_{n-1}^{f\to g})(z_{n-1}^f) \right\|_2^2 \tag{2}$$

$$\mathcal{L}_n^2 = \sum_{t,d} \left\| g_n(z_{n-1}^g) - (h_n^{f\to g} \circ f_n \circ h_{n-1}^{g\to f})(z_{n-1}^g) \right\|_2^2 \tag{3}$$

$$\mathcal{L}_n^3 = \sum_{t,d} \left\| (h_n^{f\to g} \circ f_n \circ \cdots \circ f_1)(x) - (g_n \circ \cdots \circ g_1)(x) \right\|_2^2 \tag{4}$$

Incorporating all, the loss function used in transferring the inductive bias is $\mathcal{L} = \sum_n \left(\mathcal{L}_n^1 + \mathcal{L}_n^2 + \mathcal{L}_n^3\right)$. We refer to the proposed method as block-by-block matching (BBM) because it performs knowledge distillation by matching each block of the teacher and the student.

3.2 Results

3.2.1 Details of experiments. In the ECG experiment, we divide Transformer and CNN into four blocks, respectively. Each network's blocks are divided equally, so each ResNet block contains four sub-blocks, and each transformer block contains two sub-blocks. Transformer and RNN are divided into three blocks for the EHR experiment, with each block containing one block of Transformer and one layer of LSTM, respectively.

3.2.2 Result. Table 2 shows the performance of the proposed method against other methods in Physionet 2021 and 2019. In Physionet 2021, there is a significant gap between Transformer and CNN. DeiT is unable to close this gap, but our approach not only closes the gap but also outperforms CNN. A similar trend is observed in Physionet 2019.

Table 2: Generalization performance of existing methods and proposed method. The teacher network stands for CNN in ECG dataset and RNN in EHR dataset.

	ECG		EHR	
	F-1	P-21	F-1	P-19
Transformer	0.4959	0.6070	0.1579	0.1876
Teacher	0.5890	0.6895	0.1526	0.2528
DeiT	0.5322	0.6571	0.1609	0.2177
BBM	**0.6037**	**0.7026**	**0.1610**	**0.2619**

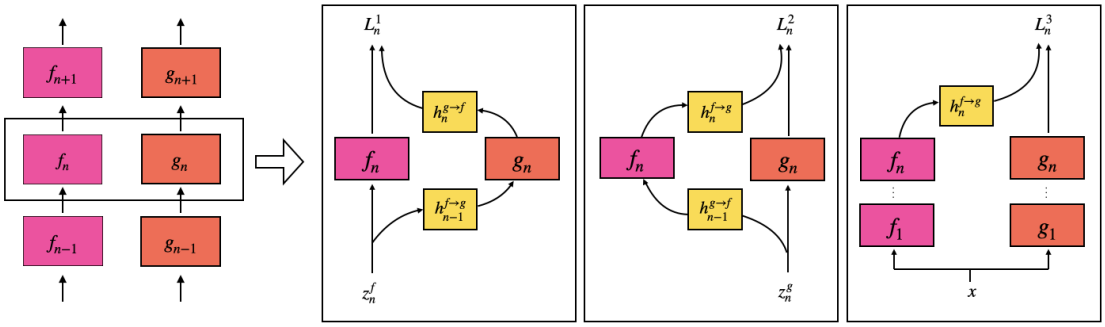

Figure 4: Pink and orange boxes are blocks of the student and the teacher, respectively. The yellow box is a dimension transformation layer. From the left to right, the panels present Equation 2, 3, and 4, respectively.

3.2.3 Evaluation on inductive bias transfer. As shown in Figure 1 and 2, BBM demonstrate higher similarity in representation with its teacher. In addition, as demonstrated in Figure 3, we observe that self-attention matrix of BBM successfully encode its teacher's inductive bias, such as spatial/temporal invariance or locality in the self-attention analysis. These prove that the proposed method encodes the inductive bias of its teacher successfully.

3.3 Ablation study

We viewed the network as composite functions, performing knowledge distillation on each function. Here, the question of the optimal number of blocks naturally arises. We perform experiments with varying the number of blocks to answer this question. As shown in Table 3, the performance increases as the number of blocks increases.

Table 3: Ablation study on the effect of the number of blocks. Each row indicate the number of blocks used for function matching. Block 1 uses the entire encoder as the single block.

	ECG		EHR	
	F-1	P-21	F-1	P-19
BBM-block1	0.5938	0.7001	0.1617	0.2066
BBM-block2	0.6027	0.7015	-	-
BBM-block3	-	-	0.1610	0.2619
BBM-block4	0.6037	0.7026	-	-

4 CONCLUSION

We show the limitation of DeiT on the transfer of inductive bias and demonstrate that this issue can be resolved by using feature-based knowledge distillation. Through experimental studies in medical data, we demonstrate that our method consistently outperforms existing methods as well as the strong inductive bias models. Additionally, an extensive analysis verifies that the proposed method transfers meaningful inductive bias to transformers. The many studies focus on transferring the inductive bias into Transformer on ImageNet. However, there are insufficient analysis on other real-world data with different properties to ImageNet. We expect our study will help bridge the gap between research on ImageNet and the real-world data.

REFERENCES

[1] Samira Abnar, Mostafa Dehghani, and Willem Zuidema. 2020. Transferring inductive biases through knowledge distillation. *arXiv preprint arXiv:2006.00555* (2020).

[2] Alibaba-Cloud. 2019. *Hefei High-tech Cup, ECG Human-Machine Intelligence Competition-Prediction of abnormal ECG events.* https://tianchi.aliyun.com/competition/entrance/231754/introduction

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 702–703.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[5] Anirudh Goyal and Yoshua Bengio. 2020. Inductive biases for deep learning of higher-level cognition. *arXiv preprint arXiv:2011.15091* (2020).

[6] Awni Y Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H Tison, Codie Bourn, Mintu P Turakhia, and Andrew Y Ng. 2019. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine* 25, 1 (2019), 65–69.

[7] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International Conference on Machine Learning*. PMLR, 3519–3529.

[8] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, Ben Recht, and Ameet Talwalkar. 2018. Massively parallel hyperparameter tuning. (2018).

[9] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. 2018. Ray: A distributed framework for emerging {AI} applications. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*. 561–577.

[10] Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. 2018. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Scientific data* 5, 1 (2018), 1–13.

[11] Matthew A Reyna, Chris Josef, Salman Seyedi, Russell Jeter, Supreeth P Shashikumar, M Brandon Westover, Ashish Sharma, Shamim Nemati, and Gari D Clifford. 2019. Early prediction of sepsis from clinical data: the PhysioNet/Computing in Cardiology Challenge 2019. In *2019 Computing in Cardiology (CinC)*. IEEE, Page–1.

[12] Matthew A Reyna, Nadi Sadr, Erick A Perez Alday, Annie Gu, Amit J Shah, Chad Robichaux, Ali Bahrami Rad, Andoni Elola, Salman Seyedi, Sardar Ansari, et al. 2021. Will Two Do? Varying Dimensions in Electrocardiography: The PhysioNet/Computing in Cardiology Challenge 2021. *Computing in Cardiology* 48 (2021), 1–4.

[13] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*. PMLR, 10347–10357.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[15] Jin Wang, Bo Peng, and Xuejie Zhang. 2018. Using a stacked residual LSTM model for sentiment intensity prediction. *Neurocomputing* 322 (2018), 93–101.