HISTOKT: CROSS KNOWLEDGE TRANSFER IN COMPUTATIONAL PATHOLOGY

Ryan Zhang[1], Jiadai Zhu[1*], Stephen Yang[1*], Mahdi S. Hosseini[2*], Angelo Genovese[3], Lina Chen[4], Corwyn Rowsell[5], Savvas Damaskinos[6], Sonal Varma[7], Konstantinos N. Plataniotis[1]*

[1]University of Toronto, Canada [2]University of New Brunswick, Canada
[3]Università degli Studi di Milano, Italy [4]Sunnybrook Health Sciences Centre, Canada
[5]St. Michaels Hospital, Canada [6]Huron Digital Pathology, Canada [7]Kingston Health Sciences Center, Canada

https://github.com/mahdihosseini/HistoKT

ABSTRACT

The lack of well-annotated datasets in computational pathology (CPath) obstructs the application of deep learning techniques for classifying medical images. Many CPath workflows involve transferring learned knowledge between various image domains through transfer learning. Currently, most transfer learning research follows a model-centric approach, tuning network parameters to improve transfer results over few datasets. In this paper, we take a data-centric approach to the transfer learning problem and examine the existence of generalizable knowledge between histopathological datasets. First, we create a standardization workflow for aggregating existing histopathological data. We then measure inter-domain knowledge by training ResNet18 models across multiple histopathological datasets, and cross-transferring between them to determine the quantity and quality of innate shared knowledge. Additionally, we use weight distillation to share knowledge between models without additional training. We find that hard to learn, multi-class datasets benefit most from pretraining, and a two stage learning framework incorporating a large source domain such as ImageNet allows for better utilization of smaller datasets. Furthermore, we find that weight distillation enables models trained on purely histopathological features to outperform models using external natural image data.

1. INTRODUCTION

Currently in the United States, there are a reported 3.94 pathologists per $100,000$ people. In Canada, this number rises slightly to 4.81 pathologists per $100,000$ people [1]. This severe scarcity of pathologists, combined with a rigorous set of duties that involves patient care and extraneous specimen diagnoses, results in decreased diagnosis quality and diminished patient experience [1]. To alleviate these burdens, the computational pathology (CPath) field has created numerous computer-aided diagnosis (CAD) tools to assist pathologist diagnoses [2]. These CAD tools utilize computer vision techniques and neural network architectures to solve a plethora of tasks, including classification, segmentation, and localization [2].

However, challenges in implementing CAD systems arise in part due to limitations in pathology datasets. Namely, despite the presence of large pathology datasets, the lack of proper annotations or labels hinders development of supervised neural networks [2]. Additionally, different standards for staining histology slides and varying optical configurations introduce further complications when creating CAD systems [3, 4, 5]. Combined, these factors result in a valuable data landscape comprised of sparse and non-comprehensive datasets.

Transfer learning is widely used in machine learning to compensate for the absence of comprehensive annotated datasets. Using transfer learning, knowledge gained from one source domain can be applied to problems on another target domain. In CPath, transfer learning using either natural image datasets or other pathology datasets enables networks to generalize to specific target domains where labeled data is scarce [6, 7, 8, 9].

The merits of transfer learning in CPath are clear: models achieve higher metrics on a target domain when pretrained on a relevant source domain dataset [6, 7, 8, 9]. These benefits are explored in previous works through model-centric approaches. While these approaches are useful in exploring quality in various model architectures, they ignore key issues introduced when these models are applied to other datasets. Due to a lack of standardized data preparation when transferring knowledge between datasets, models trained on two distinct datasets are likely to learn at two distinct biological scales. Previous work has shown that models trained from a dataset gathered by one pathology laboratory can underperform when applied to images gathered by a separate laboratory [5]. Furthermore, dataset choice for the source domain is not well explored, with little known about what makes a "good" dataset. These issues can only be resolved through a data-centric approach towards exploring the interactions of source domain data with target domain data under transfer learning.

In this paper we introduce the following contributions: *i)* we propose a standardized platform to aggregate learned histopathological knowledge, including an image standardization workflow, training, and a tuning pipeline ; *ii)* we examine the potential for aggregation of learned knowledge between multiple pathological datasets. Using cross transfer over nine classification task datasets, we evaluate both the quantity and quality of transferable information between datasets; *iii)* we propose weight distillation: a method for combining learned information from encoders trained on separate datasets. *iv)* we assess the utility of large natural image domains (ImageNet) as a source domain with two stage transfer learning; *v)* we visualize the transferred knowledge using t-SNE plots and Grad-CAM images.

2. METHODS

We introduce our pipeline for knowledge transfer, which includes dataset preprocessing, model training, and evaluation. In our evaluation, we considered the databases summarized in Tab. 1. The overall HistoKT workflow is summarized in Fig. 1.

2.1. Preprocessing

Datasets were standardized according to our pipeline, consisting of rescaling, cropping, and reflection wrapping to match the benchmark dataset, ADP. ADP was chosen as a benchmark due to its coverage of various histological tissue types.

Each image in a given dataset was rescaled to the common pixel resolution of 1 μm using the `scikit-image` library. If the resultant image is larger than 272 pixels in either dimension, the image is cropped into 272×272 patches, with 50% overlap in either direction. If the rescaled image is smaller than 272 pixels in either dimension, the image is reflection wrapped. After cropping, background images were filtered; images that had low contrast, with pixels falling

*Equal contribution

Table 1: Dataset Information

Original Dataset Information									Number of Extracted Patches		
Dataset Name	**Tissue Type**	**Diagnostic**	**Staining**	**Scanner**	**Classes**	**Dataset Size**	**Image Size**	**Pixel Resolution**	**Training Images**	**Validation Images**	**Test Images**
ADP [10]	Multi-organ	Histology (healthy)	H&E	Huron TissueScope LE1.2	33	17668	272×272	$1\ \mu m$	14134	1767	1767
MHIST [11]	Colorectal polyps	Cancer	H&E	Aperio AT2	2	3152	224×224	$1.25\ \mu m$	1740	435	977
BACH [12]	Breast	Cancer	H&E	Leica ICC50 HD	4	400	2048×1536	$0.42\ \mu m$	958	240	1199
AJ-Lymph [13]	Lymph nodes	Lymphoma	H&E	N/A	3	374	1388×1040	$0.25\ \mu m$	299	37	38
PCam [14]	Lymph nodes	Lymphoma	H&E	Pannoramic 250 Flash II, NanoZoomer-XR Digital slide scanner C12000-01	2	294912	96×96	$0.972\ \mu m$	2000	400	32322
CRC [15]	Colon & rectum	Histopathology	H&E	Online	7	107000	224×224	$0.5\ \mu m$	14000	1750	1750
GlaS [16]	Intestinal glands	Cancer	H&E	Zeiss MIRAX MIDI Slide Scanner	2	165	Various	$0.62\ \mu m$	163	112	40
OS [17]	Bone	Osteosarcoma	H&E	N/A	3	1144	1024×1024	$1\ \mu m$	13227	1653	1654
BCSS [18]	Breast	Histopathology	H&E	Online	10	151	Various	$0.25\ \mu m$	14288	1786	1787

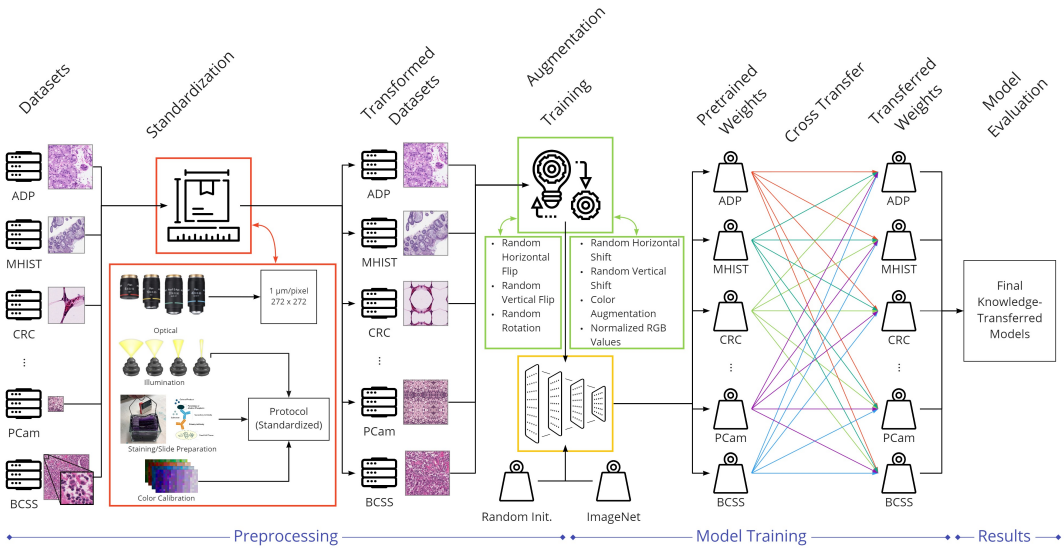


Fig. 1: In order to use a new dataset as a source domain for transfer learning, it must first be preprocessed via standardization to the ADP dataset format. Afterwards, model training and tuning enables knowledge cross transfer to a task-specific target domain. This cross transfer is carried out as tuning of a source domain trained encoder. We tune all datasets on all source datasets to evaluate performance.

between the 5th percentile and 99th percentile having less than 5% coverage of the colour span $(0 - 255)$, were removed.

This pipeline was chosen for maintaining biological scale across datasets, so that models trained on one dataset operate at the same scale when applied to another dataset.

2.2. Training

ResNet18 [19] was chosen as the baseline model due to its wide use throughout literature, as well as its relatively small number of parameters compared to other commonly selected networks.

For all experiments, training was conducted using `PyTorch`, utilizing NVIDIA Tesla V100 Tensor Core GPUs. For all baseline results, models were trained from random initialization on a given dataset, using the RMSGD optimizer [20], with an initial learning rate of 0.03, momentum of 0.9, weight decay of $5e-4$, and all other parameters left as default [21]. Multi-labeled datasets, ADP and BCSS, are trained with a weighted one-vs-all cross entropy loss, while all other datasets are trained with cross entropy loss. Baseline models were trained for 250 epochs, with three trials for each dataset. The model with the highest validation accuracy for a given epoch was taken as the baseline weight for further evaluation.

2.3. Tuning

Three primary methods were tested for tuning on a target domain: no tuning, fine-tuning, and deep-tuning. For no tuning, we take the encoder trained on a source domain and evaluate the encoder on the same domain. We denote fine-tuning to be tuning with all layers frozen except the final fully connected (FC) layer, and deep-tuning as tuning where no layers are frozen.

Our tuning procedure uses the AdamP optimizer [22], with weight decay set to $5e-4$, and all other parameters left as default. Learning rates were determined through a grid search. A learning rate scheduler was used which reduced the learning rate by a factor of two every 20 epochs. Models were trained for 250 epochs, and three trials were run for every learning rate and target domain combination.

2.4. Transferability

Transferability is evaluated using comparison matrices, as shown in Tab. 2 and Tab. 3. Along the diagonals are the average top-1 test accuracies for each dataset trained from random initialization using the methodology described in the training section. We then pick the best baseline models with respect to top-1 test accuracy as the candidate model to perform tuning, and present the average test accuracy on the target datasets. All of the (off-diagonal) results are deep-tuned, as deep-tuning greatly outperforms fine-tuning in most datasets, as shown in Tab. 4. Deep-tuning also allows us to compare the difference in learned representations with t-SNE plots, as fine-tuning does not change the encoder weights.

Table 2: HistoKT Matrix (Top-1 Test Accuracy)

Source \ Target	ADP	MHIST	BACH	AJ-LYMPH	PCam	CRC	GlaS	OS	BCSS
ADP	$93.56_{0.4}$	$78.74_{1.39}$	$93.11_{0.05}$	$94.74_{2.63}$	$76.37_{0.92}$	$99.3_{0.12}$	$88.33_{7.22}$	$94.92_{0.53}$	$97.57_{0.05}$
MHIST	$93.35_{0.07}$	$80.83_{1.41}$	$84.35_{0.91}$	$91.23_{4.02}$	$77.68_{0.61}$	$98.93_{0.12}$	$80.83_{3.82}$	$93.11_{0.37}$	$97.55_{0.04}$
BACH	$93.52_{0.03}$	$77.35_{0.43}$	$90.44_{0.85}$	$90.35_{5.48}$	$79.36_{0.66}$	$98.82_{0.07}$	$77.5_{10.9}$	$93.71_{0.31}$	$97.56_{0.09}$
AJ	$93.37_{0.02}$	$75.54_{0.99}$	$81.51_{0.19}$	$87.72_{4.02}$	$78.34_{0.9}$	$98.65_{0.17}$	$76.67_{6.29}$	$92.76_{0.36}$	$97.39_{0.07}$
PCam	$93.62_{0.04}$	$76.22_{1.32}$	$86.54_{1.21}$	$85.96_{1.52}$	$78.41_{1.39}$	$98.67_{0.26}$	$82.59_{0.01}$	$92.54_{0.21}$	$97.5_{0.03}$
CRC	$94.22_{0.04}$	$79.8_{1.13}$	$94.16_{0.58}$	$97.37_{0.0}$	$78.39_{2.34}$	$99.03_{0.06}$	$92.5_{0.0}$	$94.58_{0.07}$	$97.44_{0.06}$
GlaS	$93.29_{0.09}$	$78.68_{1.59}$	$82.85_{1.26}$	$86.84_{2.63}$	$76.66_{0.95}$	$98.9_{0.26}$	$85.0_{5.0}$	$93.07_{0.65}$	$97.48_{0.04}$
OS	$94.27_{0.12}$	$77.99_{0.74}$	$93.72_{0.24}$	$93.86_{3.04}$	$77.67_{2.11}$	$99.2_{0.23}$	$90.83_{2.89}$	$94.74_{0.3}$	$97.53_{0.02}$
BCSS	$94.03_{0.01}$	$81.88_{0.67}$	$93.61_{0.42}$	$93.86_{4.02}$	$78.52_{0.27}$	$99.05_{0.14}$	$90.0_{4.33}$	$94.56_{0.06}$	$97.67_{0.05}$

Table 3: HistoKT Matrix Pretrained on ImageNet (Top-1 Test Accuracy)

Source \ Target	ADP	MHIST	BACH	AJ-LYMPH	PCam	CRC	GlaS	OS	BCSS
ADP	$93.03_{0.28}$	$76.42_{0.21}$	$91.63_{0.88}$	$96.49_{1.52}$	$77.36_{1.42}$	$99.12_{0.18}$	$86.67_{3.82}$	$93.71_{0.44}$	$97.44_{0.03}$
MHIST	$94.47_{0.08}$	$83.52_{1.34}$	$93.47_{0.69}$	$93.86_{1.52}$	$82.57_{0.8}$	$99.16_{0.07}$	$92.5_{2.5}$	$94.1_{0.24}$	$97.73_{0.02}$
BACH	$94.4_{0.06}$	$84.27_{0.6}$	$94.41_{1.13}$	$92.98_{1.52}$	$81.63_{1.52}$	$99.26_{0.06}$	$93.33_{1.44}$	$93.93_{0.09}$	$97.73_{0.01}$
AJ	$94.15_{0.06}$	$80.08_{0.77}$	$90.6_{0.61}$	$92.11_{0.0}$	$80.0_{2.28}$	$99.12_{0.03}$	$81.67_{3.82}$	$94.32_{0.06}$	$97.68_{0.08}$
PCam	$94.41_{0.06}$	$82.09_{1.95}$	$90.96_{0.93}$	$92.98_{1.52}$	$78.67_{2.57}$	$99.26_{0.06}$	$91.67_{1.44}$	$94.1_{0.18}$	$97.62_{0.09}$
CRC	$94.38_{0.03}$	$81.58_{2.16}$	$94.61_{0.82}$	$89.47_{9.12}$	$78.27_{1.37}$	$99.35_{0.09}$	$90.83_{2.89}$	$94.94_{0.78}$	$97.51_{0.03}$
GlaS	$94.5_{0.03}$	$85.36_{1.85}$	$93.55_{0.17}$	$95.61_{1.52}$	$84.41_{1.81}$	$99.03_{0.2}$	$93.33_{2.89}$	$94.42_{0.45}$	$97.74_{0.1}$
OS	$94.12_{0.07}$	$81.03_{0.36}$	$93.44_{0.13}$	$92.11_{0.0}$	$78.86_{2.88}$	$99.07_{0.13}$	$80.83_{11.81}$	$94.2_{0.53}$	$97.52_{0.1}$
BCSS	$94.08_{0.09}$	$78.61_{1.62}$	$94.25_{0.29}$	$94.74_{2.63}$	$78.16_{1.08}$	$99.33_{0.03}$	$94.17_{2.89}$	$94.18_{0.42}$	$97.75_{0.05}$

2.5. Weight Distillation

Taking the baseline weights from top performing models, we perform weight distillation as a secondary method for evaluating the potential for knowledge aggregation. Since all of our models use the same architecture—ResNet18—each model differs only by the dataset it is trained on. For each layer of the model, we unfold the 4-D weight tensors $\mathbf{W}^l_{dataset} \in \mathbb{R}^{w \times h \times n_i \times n_o}$, where l is the layer number, into 2-D weight tensors $\overline{\mathbf{W}}^l_{dataset} \in \mathbb{R}^{n_o \times k}$, $k = w \times h \times n_i$. To combine weights from models trained on different datasets, we stack the unfolded weight tensors from all source datasets on top of each other to create a new weight tensor. For example, a new weight tensor for the 5^{th} convolutional layer

$$\overline{\mathbf{W}}^5_C = \left[\overline{\mathbf{W}}^5_{ADP}, \overline{\mathbf{W}}^5_{CRC}\right]^T$$

is created using $\overline{\mathbf{W}}^5_{ADP}$ from layer five of a model trained from random initialization on ADP and $\overline{\mathbf{W}}^5_{CRC}$ from layer five of a model trained from random initialization on CRC. We then apply Singular Value Decomposition (SVD)

$$\overline{\mathbf{W}}^l_C = \overline{\mathbf{U}}^l_C \overline{\mathbf{\Lambda}}^l_C \overline{\mathbf{V}}^{\mathbf{l}}_{\mathbf{C}}{}^T$$

to create factorized matrices. We take the first n_o rows of $\overline{\mathbf{\Lambda}}^l_C$ and the first n_o rows and n_o columns of $\overline{\mathbf{U}}^l_C$ to keep only the most important filter values in the combined weight tensor, i.e. $\overline{\mathbf{\Lambda}}^{l'} = \overline{\mathbf{\Lambda}}^l_C[:\ n_o, :]$ and $\overline{\mathbf{U}}^{l'} = \overline{\mathbf{U}}^l_C[:\ n_o, :\ n_o]$. In this way, $\overline{\mathbf{W}}^{l'} = \overline{\mathbf{U}}^{l'} \overline{\mathbf{\Lambda}}^{l'} \overline{\mathbf{V}}^l_C{}^T \in \mathbb{R}^{n_o \times k}$. Then, we fold $\overline{\mathbf{W}}^{l'}$ back to a 4-D tensor $\mathbf{W}^{l'} \in \mathbb{R}^{w \times h \times n_i \times n_o}$. For non-convolutional 1-D layers, which include batch normalization and linear layers, the resultant vector $\mathbf{w}^{l'} \in \mathbb{R}^n$ is the mean of the corresponding vectors in all input models. For 2-D linear weights, the same SVD process is carried out. The resulting model is deep tuned according to our tuning methodology.

2.6. Evaluation

We train our models on the training set and use the validation set to select the best performing model for each 250 epoch trial. Based on the validation accuracy, a single best performing model is selected to be evaluated on the test set. All results reported in this paper are test set metrics averaged over three runs. To evaluate the model on the test set, we calculate the test accuracy. All results are summarized in the results section.

Moreover, we use t-Distributed Stochastic Neighbor Embedding (t-SNE) [23] and Grad-CAM [24] to visualize our results. t-SNE visualizes high-dimensional data by giving each datapoint a location in a 2-D map. We also use Grad-CAM to visualize activation heatmaps of the network on various classes, where a warmer colour intensity corresponds to the amount of influence an image region has on a model prediction.

3. RESULTS

Single Stage Transfer Results. Tab. 2 shows the results with a single stage transfer learning process. The models with the highest top-1 test accuracies, shown on the main diagonal, are used to deep tune on the applied dataset. If a deep tuned model outperforms the baseline, the result is highlighted green, and if it underperforms compared to the baseline, the result is highlighted in red.

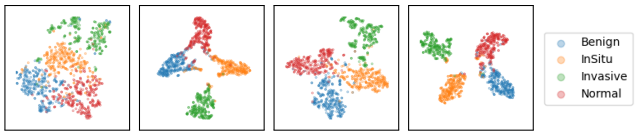
From the results, noticeable improvement from the baseline is seen when large datasets are used as a source domain: ADP, CRC, OS, and BCSS all display the ability to transfer knowledge to other, usually smaller, datasets. Training on smaller datasets like GlaS or AJ-Lymph consistently decreased performance compared to the baseline. Of the large source domains, ADP performs poorly in tasks focused on cancer detection, such as MHIST and PCam, likely due to these task being out of domain. Note that ADP is focused only on healthy tissue, while CRC, OS, and BCSS all have diseased classes. These results show that proper choice of a source domain can affect performance, and consideration of what classes in the source domain are shared with the target domain is vital.

Two Stage Transfer Results. Tab. 3 shows the top-1 test accuracy of all cross transferred datasets with a two-stage deep tuning process. Pretrained ImageNet weights are deep tuned using each dataset in the trained column, shown along the main diagonal, and these models are then deep tuned again with the applied dataset.

ImageNet pretraining improves overall performance for most datasets, and produces a higher peak performance in all datasets except AJ-Lymph, which is still within margin of error. Interestingly, datasets that negatively impacted most results in single stage transfer learning, such as GlaS, BACH, and AJ-Lymph, had many more positive interactions. We posit that this is due to ImageNet pretraining providing necessary low level features that were hard for models to learn from random initialization, due to the small size of the source domain dataset. Datasets that are multi-class and are initially hard to learn (low baseline top-1 accuracy) appear to benefit most from

Table 4: Deep-tuning vs Fine-tuning Using ADP Pretrained Weights

Tuning \ Applied	MHIST	BACH	AJ-LYMPH	PCam	CRC	GlaS	Osteosarcoma	BCSS
Deep-tuning	$78.03_{2.20}$	$93.63_{0.46}$	$95.61_{4.02}$	$76.42_{1.16}$	$99.16_{0.07}$	$86.67_{5.20}$	$94.52_{0.15}$	$97.57_{0.04}$
Fine-tuning	$69.53_{1.75}$	$65.30_{0.22}$	$71.93_{5.48}$	$77.04_{0.18}$	$80.92_{0.82}$	$80.83_{3.82}$	$82.30_{0.37}$	$87.75_{0.07}$



(a) (b) (c) (d)

Fig. 2: t-SNE plots of test sets: (a) BACH baseline; (b) BACH deep-tuned on CRC; (c) BACH deep-tuned on ImageNet; (d) BACH deep-tuned on CRC deep-tuned on ImageNet.

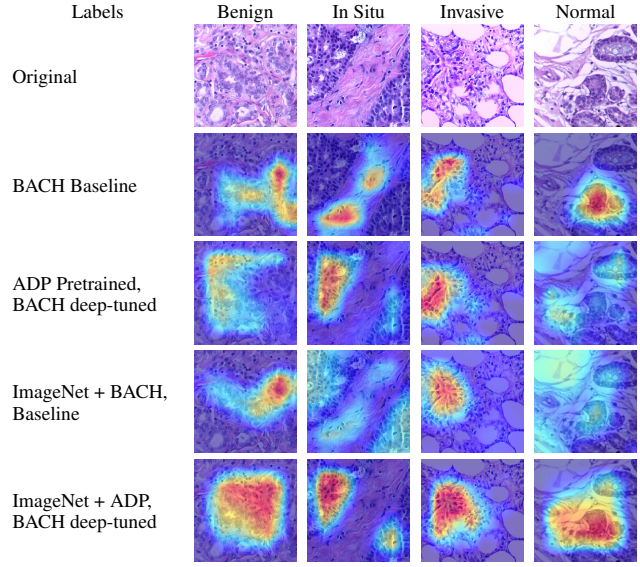


Fig. 3: Grad-CAM images for the BACH dataset.

a transfer learning process, as shown in datasets such as ADP and AJ-Lymph.

The above results describe a preliminary analysis of how much knowledge is transferable between histopathological datasets. An increased accuracy could be obtained with an experimental search of the optimal neural architecture and hyperparameters for each case. However, such analysis would be outside the scope of the paper.

t-SNE and Grad-CAM Analysis. We chose to show t-SNE results to visualize changes to latent space representation, along with Grad-CAM to show an intuitive view on how a deep-tuned model classifies images. In Fig. 2, t-SNE plots for training and test sets for the BACH dataset are shown. Deep-tuning on the best performing dataset, CRC, we see that the representation for the test set becomes more disentangled for both single stage and two stage ImageNet training procedures.

Using Grad-CAM, visualizations of the baseline model and transfer learning models are shown in Fig. 3. During generation of Grad-CAM visualizations, both augmentation smoothing and eigen smoothing were used. All activations shown are of the model output prediction, which matched with ground truth for all classes. According to expert pathologists, the ADP source domain model initialized with ImageNet weights and tuned on BACH had the best correlating activation heat map with ground truth for Benign, Invasive, and Normal classes. In contrast, the BACH model initialized with ImageNet weights (baseline model) had poor heatmap correlation with ground truth. However, both the ADP source domain model and BACH baseline model yielded the same, correct predictions for Benign, Invasive, and Normal classes.

Weight Distillation. Fig. 4 displays top-1 accuracies of top performing models evaluated on the BACH dataset. All single dataset

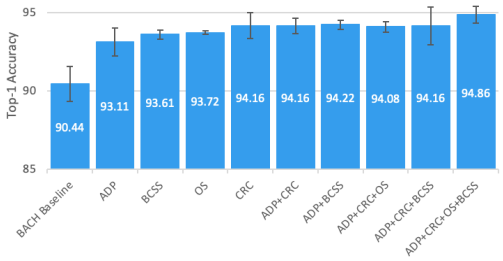


Fig. 4: Different combinations of the top performing models on the BACH dataset. The models with multiple dataset labels use weight distillation for combining weights.

results, other than the baseline, are first trained on the specified dataset, and deep tuned on the BACH dataset. For the models created by weight distillation, we first apply our weight distillation workflow to combine two or more baseline weights (no tuning), and then deep tune the resulting combined model on BACH. T-SNE and Grad-CAM visualizations for weight distilled models can be found in the supplementary material.

Here, all models outperformed the BACH baseline, and all weight distilled models perform better than or equal to the one stage deep-tuning models, other than ADP + CRC + OS. We posit that this effect is due to different CPath datasets offering knowledge from different domains, shown through our Grad-CAM analysis where models trained on different source domains learned different approaches to the same task. This enables our weight distillation model to outperform even the highest performing ImageNet tuned model, demonstrating that CPath datasets hold valuable domain specific knowledge that cannot be seen in natural image datasets.

4. CONCLUSION

In this work, we proposed and tested a cross domain knowledge transfer pipeline consisting of dataset standardization, data augmentation, and training procedures over nine histopathological datasets. To assess transferred knowledge, we conducted experiments comparing source domain viability for each of the nine datasets and two stage transfer viability using ImageNet pretrained weights. To demonstrate the validity of our transfer learning framework and to visualize the learned knowledge from one dataset to another, we use t-SNE and Grad-CAM to show the change in latent space representation and class activations, respectively. Additionally, we apply weight distillation to top performing models to aggregate knowledge across datasets. We find that knowledge is transferred between histopathological datasets, and that hard to learn, multi-class datasets benefit most from transfer learning. Datasets that share a common organ class or common tasks tend to also share knowledge more effectively, especially when the constraint of learning low level features, governed by dataset size, is removed through ImageNet pretraining. These effects are also displayed through the t-SNE and Grad-CAM analysis, with more disentangled latent representations and more meaningful class activations, respectively. Weight distillation harnesses the different learned approaches by models trained on different source domains, allowing combined models to reach higher than ImageNet pretraining accuracies, with much less computational cost compared to training on ImageNet. We present these finding in an effort to push for a more data-driven approach to transfer learning in CPath, and to create a future where CPath knowledge can be shared between any number of datasets.

5. REFERENCES

[1] D. M. Metter, T. J. Colgan, S. T. Leung, C. F. Timmons, and J. Y. Park, "Trends in the US and Canadian Pathologist Workforces From 2007 to 2017," *JAMA Network Open*, vol. 2, no. 5, pp. e194337–e194337, 05 2019.

[2] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sanchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[3] S. Otálora, M. Atzori, V. Andrearczyk, A. Khan, and H. Müller, "Staining invariant features for improving generalization of deep convolutional neural networks in computational pathology," *Frontiers in Bioengineering and Biotechnology*, vol. 7, pp. 198, 2019.

[4] M. Cui and D. Zhang, "Artificial intelligence and computational pathology," *Laboratory Investigation*, vol. 101, 01 2021.

[5] D. Tellez, G. Litjens, P. Bándi, W. Bulten, J.-M. Bokhorst, F. Ciompi, and J. van der Laak, "Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology," *Medical Image Analysis*, vol. 58, pp. 101544, Dec 2019.

[6] J. d. Matos, A. d. S. Britto, L. E. S. Oliveira, and A. L. Koerich, "Double transfer learning for breast cancer histopathologic image classification," in *Proc. of IJCNN*, 2019, pp. 1–8.

[7] J. Qu, N. Hiruta, K. Terai, H. Nosato, M. Murakawa, and H. Sakanashi, "Stepwise transfer of domain knowledge for computer-aided diagnosis in pathology using deep neural networks," in *Biomedical Engineering Systems and Technologies*, A. Roque, A. Tomczyk, E. De Maria, F. Putze, R. Moucek, A. Fred, and H. Gamboa, Eds., Cham, 2020, pp. 105–119, Springer International Publishing.

[8] M. S. Hosseini, L. Chan, W. Huang, Y. Wang, D. Hasan, C. Rowsell, S. Damaskinos, and K. N. Plataniotis, "On transferability of histological tissue labels in computational pathology," in *Proc. of ECCV*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham, 2020, pp. 453–469, Springer International Publishing.

[9] Y. Kim, S. Kim, C. Cho, I. Song, H. J. Lee, S. Ahn, S. Park, G. Gong, and N. Kim, "Effectiveness of transfer learning for enhancing tumor classification with a convolutional neural network on frozen sections," *Scientific Reports*, vol. 10, 12 2020.

[10] M. S. Hosseini, L. Chan, G. Tse, M. Tang, J. Deng, S. Norouzi, C. Rowsell, K. N. Plataniotis, and S. Damaskinos, "Atlas of Digital Pathology: A generalized hierarchical histological tissue type-annotated database for deep learning," in *Proc. of CVPR*, 2019, pp. 11739–11748.

[11] J. Wei, A. Suriawinata, B. Ren, X. Liu, M. Lisovsky, L. Vaickus, C. Brown, M. Baker, N. Tomita, L. Torresani, J. Wei, and S. Hassanpour, "A petri dish for histopathology image analysis," in *Proc. of AIME*, 2021.

[12] G. Aresta, T. Araújo, S. Kwok, S. S. Chennamsetty, M. Safwan, V. Alex, B. Marami, M. Prastawa, M. Chan, M. Donovan, G. Fernandez, J. Zeineh, M. Kohl, C. Walz, F. Ludwig, S. Braunewell, M. Baust, Q. D. Vu, M. N. N. To, E. Kim, J. T. Kwak, S. Galal, V. Sanchez-Freire, N. Brancati, M. Frucci, D. Riccio, Y. Wang, L. Sun, K. Ma, J. Fang, I. Kone, L. Boulmane, A. Campilho, C. Eloy, A. Polónia, and P. Aguiar, "BACH: Grand challenge on breast cancer histology images," *Medical Image Analysis*, vol. 56, pp. 122–139, 2019.

[13] A. Janowczyk and A. Madabhushi, "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases," *Journal of Pathology Informatics*, vol. 7, no. 1, pp. 29, 2016.

[14] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling, "Rotation equivariant cnns for digital pathology," *CoRR*, vol. abs/1806.03962, 2018.

[15] J. N. Kather, J. Krisam, P. Charoentong, T. Luedde, E. Herpel, C.-A. Weis, T. Gaiser, A. Marx, N. A. Valous, D. Ferber, L. Jansen, C. C. Reyes-Aldasoro, I. Zörnig, D. Jäger, H. Brenner, J. Chang-Claude, M. Hoffmeister, and N. Halama, "Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study," *PLOS Medicine*, vol. 16, no. 1, pp. 1–22, 01 2019.

[16] K. Sirinukunwattana, J. P. Pluim, H. Chen, X. Qi, P.-A. Heng, Y. B. Guo, L. Y. Wang, B. J. Matuszewski, E. Bruni, U. Sanchez, A. Böhm, O. Ronneberger, B. B. Cheikh, D. Racoceanu, P. Kainz, M. Pfeiffer, M. Urschler, D. R. Snead, and N. M. Rajpoot, "Gland segmentation in colon histology images: The glas challenge contest," *Medical Image Analysis*, vol. 35, pp. 489–502, 2017.

[17] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, "The cancer imaging archive (tcia): Maintaining and operating a public information repository," *Journal of digital imaging*, vol. 26, 07 2013.

[18] M. Amgad, H. Elfandy, H. Hussein, L. A. Atteya, M. A. T. Elsebaie, L. S. Abo Elnasr, R. A. Sakr, H. S. E. Salem, A. F. Ismail, A. M. Saad, J. Ahmed, M. A. T. Elsebaie, M. Rahman, I. A. Ruhban, N. M. Elgazar, Y. Alagha, M. H. Osman, A. M. Alhusseiny, M. M. Khalaf, A.-A. F. Younes, A. Abdulkarim, D. M. Younes, A. M. Gadallah, A. M. Elkashash, S. Y. Fala, B. M. Zaki, J. Beezley, D. R. Chittajallu, D. Manthey, D. A. Gutman, and L. A. D. Cooper, "Structured crowdsourcing enables convolutional segmentation of histology images," *Bioinformatics*, vol. 35, no. 18, pp. 3461–3467, 02 2019.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of CVPR*, 2016, pp. 770–778.

[20] M. S. Hosseini and K. N. Plataniotis, "AdaS: Adaptive scheduling of stochastic gradients," *CoRR*, vol. abs/2006.06587, 2020.

[21] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. of ICML*, S. Dasgupta and D. McAllester, Eds., Atlanta, Georgia, USA, 17–19 Jun 2013, vol. 28 of *Proceedings of Machine Learning Research*, pp. 1139–1147, PMLR.

[22] B. Heo, S. Chun, S. J. Oh, D. Han, S. Yun, G. Kim, Y. Uh, and J.-W. Ha, "Adamp: Slowing down the slowdown for momentum optimizers on scale-invariant weights," in *Proc. of ICLR*, 2021.

[23] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, Oct 2019.