Memory-Augmented Capsule Network for Adaptable Lung Nodule Classification

Aryan Mobiny, Pengyu Yuan, Pietro A. Cicalese, Supratik K. Moulik, Naveen Garg, Carol C. Wu, Kelvin Wong, Stephen T. Wong, Tian Cheng He, Hien V. Nguyen.

***Abstract*— Computer-aided diagnosis (CAD) systems must constantly cope with the perpetual changes in data distribution caused by different sensing technologies, imaging protocols, and patient populations. Adapting these systems to new domains often requires significant amounts of labeled data for re-training. This process is labor-intensive and time-consuming. We propose a memory-augmented capsule network for the rapid adaptation of CAD models to new domains. It consists of a capsule network that is meant to extract feature embeddings from some high-dimensional input, and a memory-augmented task network meant to exploit its stored knowledge from the target domains. Our network is able to efficiently adapt to unseen domains using only a few annotated samples. We evaluate our method using a large-scale public lung nodule dataset (LUNA), coupled with our own collected lung nodules and incidental lung nodules datasets. When trained on the LUNA dataset, our network requires only 30 additional samples from our collected lung nodule and incidental lung nodule datasets to achieve clinically relevant performance (0.925 and 0.891 area under receiving operating characteristic curves (AUROC), respectively). This result is equivalent to using two orders of magnitude less labeled training data while achieving the same performance. We further evaluate our method by introducing heavy noise, artifacts, and adversarial attacks. Under these severe conditions, our network's AUROC remains above 0.7 while the performance of state-of-the-art approaches reduce to chance level.**

***Index Terms*— Capsule network, computer-aided diagnosis, incidental lung nodule, lung nodule, meta-learning.**

I. Introduction

LUNG cancer is consistently ranked as the leading cause of cancer-related deaths all around the world in the past several years, accounting for more than one-quarter (26%) of all cancer-related deaths [1]. The stage at which diagnosis is made largely determines the overall prognosis of the patient.

A. Mobiny, P. Yuan, P. A. Cicalese, and H. V. Nguyen are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, 77004 USA (e-mail: amobiny@uh.edu, and pyuan2@uh.edu, and pcicalese@uh.edu, and hienvnguyen@uh.edu).

S. K. Moulik is with Triradiate Industries, Sugar Land, TX, 77479 USA (e-mail: skmoulik@gmail.com)

N. Garg and C. C. Wu are with the Department of Diagnostic Radiology, University of Texas MD Anderson Cancer Center, Houston, TX, 77030 USA (e-mail: ngarg@mdanderson.org, and ccwu1@mdanderson.org)

K. Wong, and T. C. He, and S. T. Wong are with the Department of Systems Medicine and Bioengineering, Houston Methodist, Houston, TX 77030, USA (e-mail: kwong@houstonmethodist.org, THe@houstonmethodist.org, STWong@houstonmethodist.org)

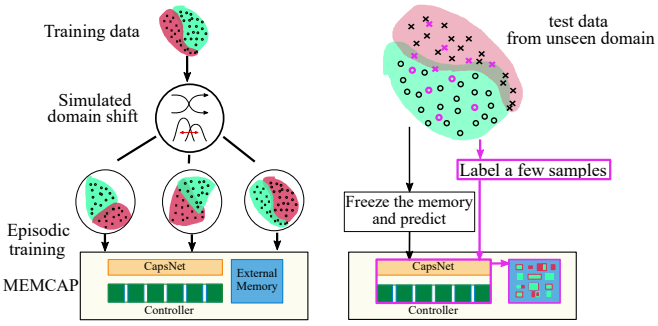

Fig. 1. Overview of the episodic meta-training (Left) and meta-testing (Right) of the MEMCAP for adaptive lung nodule classification. During the meta-training phase, the model learns to quickly encode and retrieve the required information about the new domain (task) using a small subset of annotated samples. During the meta-testing phase, the model performance is evaluated on samples from never-before-seen target domains.

The five-year relative survival rate is over 50% in early-stage disease, while survival rates drop to less than 5% for late-stage disease [1]. Lung cancer screening of high-risk individuals, which is designed to detect the disease at an early stage, has been shown in the National Lung Screening Trial (NLST) to reduce lung cancer mortality by 20% [2]. The main challenge in lung cancer screening is detecting lung nodules [2]. Radiologist fatigue, increasing workload, and stringent turn-around-time requirements are just a few of the factors which negatively impact the detection rate for lung nodules. Many studies have documented the occurrence of diagnostic errors in clinical practice, caused by many different contributing factors which can generally be divided into person-specific (e.g., satisfaction of search, etc.), nodule-specific (e.g., small size, low density) and environment-specific issues (e.g., inadequate equipment, staff shortages, excess workload, etc.) [3], [4].

Computer-aided diagnosis (CAD) systems aim to improve the radiologist's performance in terms of diagnostic accuracy and speed [5]. The role of CAD systems in lung nodule detection and screening has been demonstrated over the years [6], as well as their role in distinguishing between benign and malignant nodules [5]. However, the automated identification of nodules from non-nodules is quite challenging mainly due to the large variation in size, shape, margins, and density of the nodules [7]. The nodules can also occur in different locations (such as peri-fissural, subpleural, endobronchial, perivascular), contributing to the diversified contextual environment around

the nodule tissue [8]. In recent years, deep learning technology has attracted considerable interest in the computer vision and machine learning community [9], [10]. Deep neural networks (DNNs) have an advantage of automatically capturing the image's higher-level features directly from the raw input data. This leads to powerful features tuned to specific tasks of medical image analysis [11]–[13]. Recent work has explored deep networks for detecting lung pathology [13], [14].

Machine learning models are typically trained under the assumption that training and test data are sampled from the same distribution. This assumption is often violated as conditions for data acquisition may change, and a trained system may fail to produce accurate predictions for unseen data affected by a domain shift. In medical imaging, images acquired at different sites can differ significantly in their data distribution, due to varying scanners, imaging protocols or patient cohorts, thus each new hospital can be regarded as a new domain [15]. However, it is impractical to collect large datasets from each institution in order to update a trained system. In pulmonary nodule detection, while most of the existing work focuses on improving classification accuracy with a static dataset (i.e. the LUNA dataset [16]), the problem of adapting a classifier to changes in lung CT data is largely under-investigated. There are many inter-patient and intra-patient sources of variability in pulmonary CT scans that can affect classifier performance negatively; not only does each patient have unique characteristics, but each imaging center utilizes different CT scanners with different imaging protocols which lead to large variations in image contrast and signal-to-noise ratio. The Hounsfield Unit (HU, a measure of radiodensity) differences observed between various CT scanners due to poor calibration can also be significant, with the measured HU values of each scanner changing substantially with repeated use [17]. A possible solution is to use domain adaptation (DA) methods to narrow down the domain shift between the target and source domain [18], [19]. To the best of our knowledge, there are no established techniques for the adaptation of lung nodule classifiers to the settings unique to each hospital by using only a few labeled examples from the new domain [20], [21].

Motivated by the above challenges, this paper proposes a novel framework for the on-the-fly adaptation of a lung nodule classifier to changes in the data distribution. This eliminates the need for both expensive data annotation and training models from scratch in each of the target institutions. Fig. 1 shows an overview of the proposed framework during the training and test phases. It takes sequential labeled samples, evaluates the performance enhancement in real-time, and stops when satisfied with the model prediction performance, making it data efficient. Our paper makes the following contributions:

- We develop an efficient meta learning method, called memory-augmented capsule networks (MEMCAP), for rapidly adapting to target domains using only a few labeled samples.
- While traditional meta-learning methods require multiple tasks (each task consists of data and labels) for training, we develop a domain shift simulation strategy to train our model using only one dataset. This contribution makes meta learning technology significantly easier to use in medical applications where annotations are expensive and difficult to obtain.
- We perform extensive experiments on three different lung nodule datasets to validate the proposed network. This includes the classification of incidental pulmonary lung nodules from contrast enhanced and non-contrast enhanced CT scans acquired with normal radiation dose. These nodules were detected during screening for another disease in patients with low-risk of lung cancer. Despite the changes in the lesion size, morphology image contrast and radiation dose, our model learns to incorporate the underlying information from the target domain and makes accurate predictions for these complex cases.

II. Related Work

Domain Adaptation algorithms aim to train models that produce accurate predictions for unseen data with domain shift. Over the past 10 years, research in domain adaptation provides a number of effective techniques to mitigate the domain shift problem [22]–[30]. Donahue *et al.* [27] showed that features of a deep network generalize well to unseen domains. Nguyen *et al.* [28] proposed a network of sparse representations for adaptation on multiple levels of the feature hierarchy . In a similar spirit, Rusu *et al.* [29] proposed a progressive network, by appending the source network to the target network, enabling the architecture to reuse both low-level and high-level features. Some approaches have proposed to learn a target classifier regularized against the source classifier to facilitate adaptation when a limited amount of labeled data is available in the target domain [31], [32]. Most of these domain adaptation techniques require a computationally intensive re-training process (e.g, hours to days) to achieve a comparable prediction performance in the target domain. Such a requirement will inevitably interrupt clinical workflows and cause a significant delay in diagnoses and treatments. Moreover, due to the small numbers of data samples from new domains, the re-training process is prone to poor generalization. It is also unrealistic to expect the user to collect many labeled data in each new domain [33], [34]. In contrast, many clinical settings require real-time learning and inference from a small amount of data provided by physicians. This kind of flexible adaptation remains a major challenge for the existing approaches, including those using deep networks [35].

Meta-learning (a.k.a learning to learn) explores the training of a meta-learner that learns to learn new tasks and skills quickly with a few training samples [36], [37]. Model-agnostic meta-learning (MAML) [38] is a gradient-based procedure that incorporates an episodic training paradigm for the fast adaptation of models to new tasks and domains. Santoro et al. [39] proposed a memory-augmented neural network (a recurrent network with an explicit storage buffer) with a set of modifications on the training set up and the memory retrieval mechanism which allows it to encode new information quickly and thus adapt to new tasks after only a few samples. Bercea *et al.* [40] introduced a memory augmented network that

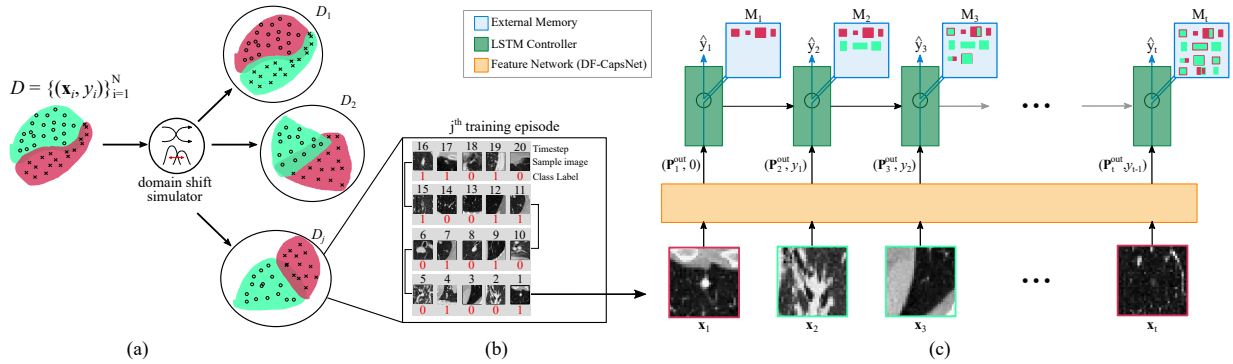

Fig. 2. Overview of the proposed model-based meta-learning approach for adaptive lung nodule classification. (a) Simulated domain shift, (b) Sampled sequence of images for the episodic training, (c) Architecture of the MEMCAP classifier, consisting of a feature network (a 3D Capsule Network named FastCaps++) and a task network (composed of an LSTM controller with an external memory bank). MEMCAP takes a few annotated inputs from the new domain and refines its decision function in real-time.

decomposes input X-Ray images into several patches that are then sequentially segmented and combined. Zhang *et al.* [41] also introduced a memory augmented adversarial network for retinal OCT screening that enhances anomalies present in the data that are difficult to detect. While the meta-learning approaches are significantly different in motivation and methodologies, their training requires the availability of multiple datasets [38], [42]. In medical applications, this requires collecting and annotating data from different institutes which is often expensive and inefficient.

III. Methodology

A. Memory-Augmented Capsule Network

The proposed MEMCAP architecture makes no assumption about the form of the output $P_\theta(y|X)$, which is the probability of a data point X belonging to the class y. We denote input and output (label) spaces by $\mathcal{X}$ and $\mathcal{Y}$ respectively. MEMCAP is composed of two sub-modules; first, a feature extractor network $F_\theta : \mathcal{X} \to \mathcal{Z}$ extracts discriminative features from the large volumetric input images, where $\mathcal{Z}$ is the feature space of much lower dimension than $\mathcal{X}$. It then passes the embedding vectors to a task network $T_\phi : \mathcal{Z} \to \mathbb{R}^C$ where C is the number of classes in $\mathcal{Y}$. The final class predictions are given by

$$\hat{y} = p(y|\mathbf{x}; \theta, \phi) = \text{softmax}(T_\phi(F_\theta(\mathbf{x}))) \tag{1}$$

where $\text{softmax}(a) = e^a / \sum_i a_i$. The parameters θ, ϕ are optimized with respect to a classification ($\mathcal{L}_{\text{class}}$) and a task-specific ($\mathcal{L}_{\text{task}}$) objective function. We elected to use a deep capsule network architecture as the feature extractor network due to its ability to extract various invariant low-dimensional properties of the entities across domains; including different types of instantiation parameters such as position, size, orientation, etc. [43]. It then passes the abstract representations to the task network: a recurrent controller with an explicit storage buffer to rapidly encode and store the new information extracted from the labeled target examples, thus exploiting the underlying intrinsic information in the target domain for prediction [39]. Finally, we propose a stand-alone approach for simulating the domain shift during training and testing by applying random distortions to input and output data. The detailed information about the model architectures, the task set up and training strategies are provided in the following sections.

B. FastCaps++ as Feature Extractor Network

Capsule networks (CapsNets) were proposed by Sabour et al. [43] as an alternative to convolutional neural networks (CNNs) that possess multiple desirable properties such as the ability to: generalize with fewer training examples, encode and compress a vast amount of information in short pose vectors and matrices, and being significantly more robust to adversarial attacks and noisy artifacts [14], [44]. These properties prompted us to use a CapsNet with slight variations to encode input instantiation parameters into lower-dimensional vectors.

A CapsNet is composed of a cascade of capsule layers, each of which contains multiple capsules. A capsule is the basic unit of CapsNets and is defined as a *group of neurons* whose output forms a *pose* vector or matrix [43], [44]. This is in contrast to traditional deep networks that use neurons as their basic unit. In this work, we elected to use matrix capsules as it helps with reducing the number of trainable parameters required by the transformation matrices [44] which eventually made our network less prone to over-fitting. Let Ω_L denote the sets of capsules in layer L. Each capsule $i \in \Omega_L$ outputs a pose matrix P_i^L. Each element in the matrix characterizes the instantiation parameters (such as orientation, size, pose, etc.). The activation probability of a capsule a_i^L indicates the presence of an entity and is implicitly encoded in the capsule as the Frobenius norm of the pose matrix. The i-th capsule in Ω_L propagates its information to j-th capsule in Ω_{L+1} through a linear transformation $\mathrm{V}_{ij}^L = \mathrm{W}_{ij}^L \mathrm{P}_i^L$, where V_{ij}^L is called a *vote* matrix. The pose matrix of capsule $j \in \Omega_{L+1}$ is a weighted combination of all the votes from child capsules: $\mathrm{P}_j^{(L+1)} = \sum_i r_{ij} \mathrm{V}_{ij}^L$, where r_{ij} are routing coefficients and $\sum_i r_{ij} = 1$. These coefficients are determined by the dynamic routing algorithm [43] which iteratively increases the routing coefficients r_{ij} if the corresponding voting matrix V_{ij}^L is similar to P_j^{L+1} and vice versa. Dynamic routing ensures that the output of each child capsule gets sent to the proper parent capsules. Through this process, the network gradually

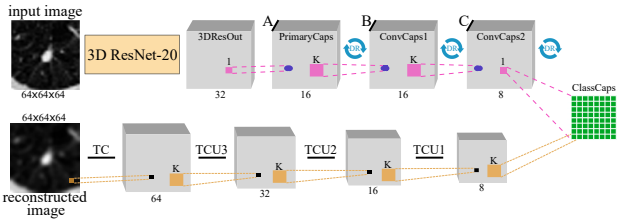

Fig. 3. Illustration of the encoder (top) and decoder (bottom) paths of the FastCaps++ architecture. It outputs an 8×8 matrix of semantic features extracted from the high-dimensional input.

constructs a transformation matrix for each capsule pair to encode the corresponding part-whole relationship and retains geometric information of the input data.

We applied a few simple yet effective modifications to the Fast CapsNet proposed by Mobiny et al. [14] to make it scale properly to our high-dimensional volumetric inputs and improve its convergence speed and prediction performance while requiring a smaller number of trainable parameters. We called our model FastCaps++ whose architecture is depicted in Fig. 3. FastCaps++ is composed of an encoder and decoder path. The encoder path uses a 3D ResNet-20 [45] with three residual blocks as the base network, followed by a 1×1 convolution layer which outputs $A = 64$ feature maps. All the other layers are capsule layers starting with the primary capsule layer. The 4×4 pose matrix of each of the B primary capsule maps is a learned linear transformation of the output of all the lower-layer ReLUs centered at that location. The primary capsules are followed by two convolutional capsule layers with C and D capsule maps and kernels of size $K = 3$ and stride $s = 1$ and $s = 2$, respectively. We selected $B = C = D = 32$ for the capsule layers, and used a dynamic routing mechanism to route the information between the capsules. The last layer of convolutional capsules is linked to the final dense capsule layer which has only one capsule with an 8×8 pose matrix. The Frobenius norm of the pose matrices of the output capsule is used to determine the predicted class (nodule vs. non-nodule).

The decoder network then reconstructs the input from the final capsules, which will force the network to preserve as much information from the input as possible across the whole network. This effectively works as a regularizer that reduces the risk of over-fitting and helps generalize to new samples. Inspired by [46] we used a convolutional architecture composed of three transposed convolution units (TCU) to reconstruct the input volume. Each TCU doubles the volume size and is composed of two 3D transposed convolution layers (with $K = 1, s = 1$ and $K = 3, s = 2$, respectively), each of which is followed by a ReLU non-linearity. The final transposed convolution layer (shown as TC in Fig. 3) uses a kernel of size $K = 1$ and stride $s = 1$ with a sigmoid non-linearity to map the values into the $[0, 1]$ range.

C. Memory-Augmented Task Network

Due to the sequential nature of annotated feedback samples provided from each new target domain, and the need to encode and accumulate the information over time, neural networks with recurrent structures are a natural choice. They are equipped with an "internal memory" that captures information about what has been calculated so far. The long-short term memory (LSTM) model is introduced as a modification to vanilla recurrent networks (RNNs), which is capable of encoding long-term dependencies [47].

In the lung nodule detection problem, the model must be able to quickly encode and retrieve information, and modify its decisions to make accurate inferences by using a few annotated samples. This set of samples are provided as either information correction by the radiologist or brand-new samples from never-before-seen distributions. Thus the ideal model must learn to capture the cumulative expertise gained *across* domains and continuously adapt to never-before-seen distributions [48]. However, neural networks with internal memory capacity (such as LSTM) are not able to rapidly encode, store and access a significant amount of new information required at each step. Architectures such as memory networks [49] and Neural Turing Machines (NTMs) are developed as models that meet the requisite criteria. These are external-memory equipped networks capable of rapidly encoding new information and storing them in a stable, addressable representation that can selectively be accessed when needed. Inspired by [39], we use an LSTM architecture equipped with an external memory bank as the task network (as shown in Fig. 2) for sequentially processing the new information. The external memory bank interacts with the LSTM controller through reading and writing operations. The external memory is denoted by a matrix $\mathbf{M}_t \in \mathbb{R}^{k \times q}$ where k is the number of memory slots and q is the size of each slot. The model has an LSTM controller that reads and writes to the external memory at every time step (i.e. receiving each new annotated sample).

Reading: For a given input $\mathbf{x}_t$ and the memory matrix $\mathbf{M}_t$ with k rows (slots) of size q at time t, the controller will produce a key $\mathbf{k}_t$ computed as $\mathbf{k}_t = tanh(\mathbf{W}_{hk}\mathbf{h}_t + \mathbf{b}_k)$ from the controller hidden states ($\mathbf{h}_t$). $\mathbf{W}_{hk}$ and $\mathbf{b}_k$ are the corresponding weight matrix and bias values respectively. This key will be compared against each memory slot $\mathbf{M}_t(i)$ using the cosine similarity measure $C(.,.)$. This similarity is used to produce the read-weight vector $\mathbf{w}_t^r$:

$$\mathbf{w}_t^r = \text{softmax}\big[C(\mathbf{k}_t, \mathbf{M}_t(i))\big] \tag{2}$$

where softmax is used to get the normalized weight vector with its elements summing to one. This vector allows the controller to select values similar to previously-seen values, which is called content-based addressing. Finally, the reading operation is done by a weighted linear combination of all memory slots scaled by a normalized read-weight vector $\mathbf{w}_t^r$ as follows:

$$\mathbf{r}_t = (\mathbf{M}_t)^T.\mathbf{w}_t^r \tag{3}$$

Here, $\mathbf{r}_t$ is the content vector retrieved from the memory, and $\mathbf{w}_t^r \in \mathbb{R}^{k \times 1}$ is the read-weight vector which specifies how much each slot should contribute to $\mathbf{r}_t$,

Writing: To write into the memory, the controller will interpolate between writing to the most recently read memory rows and writing to the *least-used* memory rows. If $\mathbf{w}_{t-1}^r$ is the read-weight vector at the previous time step, and $\mathbf{w}_{t-1}^{lu}$ is a weight vector that captures the least-used memory location, the

Algorithm 1 Episodic training with simulated domain shift.

Input: Source training domains $\mathcal{D} = \{\mathcal{D}_k\}_{k=1}^{K}$
Output: Feature extractor (F_θ) and task network (T_ϕ)

Randomly split $\mathcal{D}$ into disjoint $\mathcal{D}^{\text{tr}}$ and $\mathcal{D}^{\text{te}}$
repeat
  Randomly select a simulated domain $\mathcal{D}_k^{\text{tr}}$
  $\mathbf{M} \leftarrow 0$ ▷ Reset for each meta-training episode
  for $\mathbf{x}_t \in \mathcal{D}_k^{\text{tr}}$ **do** ▷ for each labeled sample
    $\mathbf{P}_t^{\text{out}}, a_t = F_\theta(\mathbf{x}_t)$
    $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{class}}(y_t, a_t; \theta)$ ▷ update F_θ
    compute $\mathbf{k}_t$, $\mathbf{w}_t^r$, $\mathbf{w}_t^w$, $\mathbf{r}_t$, $\mathbf{a}_t$ ▷ Section III-C
    $\mathbf{M}_t \leftarrow \mathbf{M}_{t-1} + \mathbf{w}_t^w . \mathbf{a}_t$ ▷ update the memory
    $\hat{y}_t = T_\phi((\mathbf{P}_t^{\text{out}}, y_{t-1}); \mathbf{r}_t, \mathbf{h}_t)$
    $l \leftarrow l + \mathcal{L}_{\text{task}}(y_t, \hat{y}_t; \theta, \phi)$
  $(\theta, \phi) \leftarrow (\theta, \phi) - \gamma \frac{1}{|\mathcal{D}^{\text{tr}}|} \nabla l(\mathcal{D}^{\text{tr}}; \theta, \phi)$ ▷ update F_θ, T_ϕ
  $\mathbf{M} \leftarrow 0$ ▷ Reset for each meta-testing episode
  $\hat{y}_{t'} = T_\phi(F_\theta(\mathbf{x}_{t'}))$ ▷ Evaluate for $\mathbf{x}_{t'} \in \mathcal{D}_j^{\text{te}}$
  $\mathcal{L} \leftarrow \frac{1}{|\mathcal{D}^{\text{te}}|} \sum_j \sum_{t'} \mathcal{L}_{\text{task}}(y_{t'}, \hat{y}_{t'}; \theta, \phi)$
until convergence

write weights $\mathbf{w}_t^w \in \mathbb{R}^{1\times k}$ is then computed using a learnable sigmoid gate:

$$\mathbf{w}_t^w \leftarrow \sigma(\alpha_t)\mathbf{w}_{t-1}^r + (1 - \sigma(\alpha_t))\mathbf{w}_{t-1}^{lu} \tag{4}$$

where $\sigma(.)$ is the sigmoid function. α_t is a scalar computed as $\alpha_t = \mathbf{w}_\alpha \mathbf{h}_t + b_\alpha$ at each time step, and $\mathbf{w}_\alpha$ and b_α are trainable parameters learned discriminatively through back-propagation. This encourages information to be written into either rarely-used locations of the external memory to preserve recently encoded information, or the last used location to *update* the memory with newer, possibly more relevant information. The i^{th} memory slot at time-step t, $\mathbf{M}_t(i)$, is then updated as:

$$\mathbf{M}_t(i) \leftarrow \mathbf{M}_{t-1}(i) + \mathbf{w}_t^w(i).\mathbf{a}_t \tag{5}$$

where $\mathbf{a}_t$ is the linear projection of the current hidden state followed by a *tanh* nonlinearity.

To create the least used weight vector $\mathbf{w}_t^{lu}$, the controller maintains a usage-weight vector $\mathbf{w}_t^u$ which gets updated after every read and write step as:

$$\mathbf{w}_t^u \leftarrow \beta \mathbf{w}_{t-1}^u + \mathbf{w}_t^r + \mathbf{w}_t^w \tag{6}$$

where $\beta \in [0, 1]$ is a scalar parameter used to determine how quickly previous usage values should decay. The least used weight vector $\mathbf{w}_{t-1}^{lu}$ is a one-hot-encoded vector generated from $\mathbf{w}_{t-1}^u$ by setting its minimum element to 1, and all other elements to 0. Finally, the concatenation of the read content vector and the hidden nodes $(\mathbf{r}_t, \mathbf{h}_t)$ is used to predict the output. The introduction of external memory enables the recurrent network to store and retrieve much longer-term information compared to LSTM. This frees up the main controller and increases its capacity to learn highly complicated patterns within the data.

D. Episodic Training with Simulated Domain Shift

Our learning procedure is an episodic training scheme meant to expose the model optimization to distribution mismatches. The idea of episodic training is inspired by human learning and evolution through generations [38], [39]. Each training episode mimics a learner's lifespan where it learns to optimize its performance. The next episodes are like the next generation learners using the accumulated knowledge to solve a similar problem regardless of a possible shift in the data distribution. Given the labeled training data $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, we synthetically generate samples from new domains $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_K\}$ by altering the input/output data distributions. In our implementation, training on each sequence of images from the new domain $\mathcal{D}_k$ is called an *episode*.

Theoretically, any data transformation technique can be applied to slightly change the distribution of data. By increasing the diversity of the meta tasks, the model can be well trained to extract useful features after adapting to training data in any task. We elected to synthetically enhance each meta-training task by using a variety of systematic transformations that equivalently alter the associated data and labels to simulate a new domain $\mathcal{D}_k = \{(\mathbf{x}_n^{(k)}, y_n^{(k)})\}_{n=1}^{N_k}$ where N_k is the number of labeled samples in the k-th domain. This is done by randomly applying affine transformations (scale, translate, shear), jittering the pixel intensities by applying Gaussian blurring and changing the contrast and/or brightness of the input images. This effectively simulates the domain shift in a real-world medical imaging scenario where differences in CT scanners, imaging protocols, and many other factors change the distribution of pixel intensity values [50]. Moreover, labels are randomly flipped from episode to episode (during training only). For example, nodules can be labeled as 1 in one episode and 0 in another. Note that labels are consistent within the same episode. This strategy helps prevent the MEMCAP from learning a fixed mapping from samples to their class labels, but a dynamic binding between image features and the provided labels. Consider the scenario where we do not flip the labels, the network can simply predict the ground-truth labels from input images instead of relying on the provided labels. This is undesirable as the model becomes insensitive to the new information from the target domain. This is inspired by the strategies used in other studies to learn dynamic mappings for different tasks [39], [51]. Our extensive experiments show that introducing such synthetic domain shifts significantly improves the cross-domain transferability of the learned model.

Algorithm 1 summarizes the episodic training of the MEMCAP model. We first split the source domains $\mathcal{D} = \{\mathcal{D}_k\}_{k=1}^{K}$ into disjoint meta-train ($\mathcal{D}^{\text{tr}}$) and meta-test ($\mathcal{D}^{\text{te}}$) sets. In the meta-training phase, we train the system by sequentially feeding the input $\mathbf{x}_t$ from a new domain $\mathcal{D}_k^{\text{tr}}$ to FastCaps++ (feature extractor network) to extract semantic features $\mathbf{P}_t^{\text{out}} \in \mathcal{Z}$, which will then be fed along with the time-delayed output y_{t-1} to the task network to predict the current label y_t (also shown in Fig. 2).

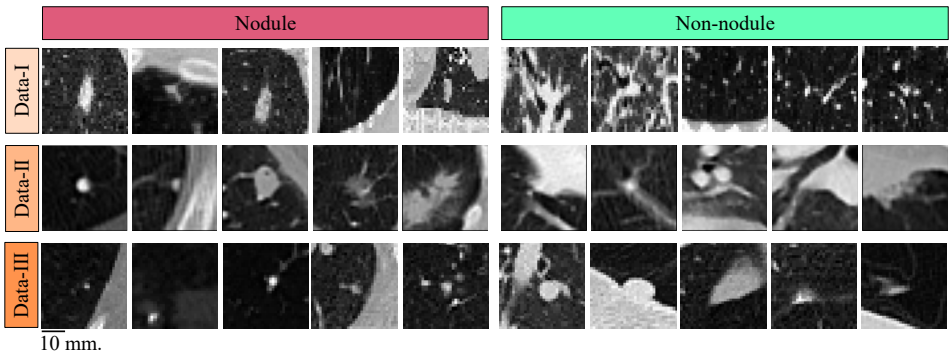

Fig. 4. Sample images of nodules (Left) and non-nodules (Right) selected from Data-I (LUNA), Data-II (our dataset), and Data-III (incidental lung nodule). Each image is a slice along the 2D axial plane in the middle of the volume. Samples from Data-II are numbered and discussed in the text.

IV. Dataset

We used three different sets of CT images in our experiments. These datasets were collected and pre-processed independently, and thus it is reasonable to consider their images as samples from different distributions. Fig. 4 shows examples of extracted candidates and the corresponding labels from all three sets. These images illustrate the highly challenging task of distinguishing nodules from non-nodule lesions as the pulmonary nodules come with large variations in shapes, sizes, types (solid, sub-pleural, cavitary, and ground-glass) and the non-nodule candidates often mimic the morphological appearance of the real pulmonary nodules.

A. Data-I: LUNA-16 Lung Nodule Dataset

We use the candidate nodules provided by the LUNA-16 challenge [16] as the source data which is used for the initial meta-training and meta-testing phases to train and evaluate the model. This dataset is a subset of LIDC-IDRI data [52], the largest publicly available reference database for lung nodules including a total of 1018 CT scans. It consists of low-dose CT scans collected from a wide range of scanner models and acquisition parameters from seven different participating academic institutions. LUNA-16 includes candidate nodules (of size $\geq$ 3mm) generated from only 888 scans and labeled by experienced thoracic radiologists in a two-phase reading process. In the first phase, four radiologists annotated scans and marked all suspicious lesions. In the next phase, the annotations generated by all other radiologists were revealed to each radiologist who then independently reviewed all annotations [16]. This results in 750K candidate nodules of size $64 \times 64 \times 64$, containing about 1500 true nodules. We withhold 20% of the data for testing and perform five fold cross validation on the remaining 80%. We evaluate the performance of the trained models on the disjoint test dataset.

B. Data-II: Colleted Lung Nodule Dataset

This dataset includes 226 unique CT Chest scans captured by General Electric and Siemens scanners at the MD Anderson Cancer Center on a single day. The data was pre-processed by the Triradiate Industries - Autonomous Image Recognition software which is a proprietary automated segmentation system trained on datasets from the same institution. The core algorithm is an expert system used primarily for isolating lung parenchyma and blood vessels. The system uses a combination of density filters and simple feature analysis to produce a voxel level segmentation map. From within the segmented lung tissue, a set of potential nodule points is generated based on the size and shape of regions within the lung which exceeds the air HU threshold. Additional filters, based on symmetry and other common morphological characteristics, are applied to decrease the false positive rate while maintaining very high sensitivity. Bounding boxes with at least 8 voxels padding surrounding the candidate nodules are cropped and resized to $64 \times 64 \times 64$ pixels. Each generated candidate is reviewed and annotated by board-certified radiologists. From all the generated images (about 7400 images), around 56% were labeled as nodules while the rest were labeled as non-nodules. The radius of the detected nodules ranges from 3 to 15 mm with an average of 5 ($\pm$1.86) mm. The differences in scan protocol for the various chest studies yield slice thickness ranging from 0.625 - 2.5 mm. In each case, an attempt was made to get as close to isotropic as possible while adhering to slice thickness multiples of 0.625mm. For example, most studies have an in-plane resolution of approximately 0.7×0.7 mm; in this case, linear interpolation was used to convert the data to 0.625mm slice thickness (voxel dimension of $0.7 \times 0.7 \times 0.625$mm).

C. Data-III: Collected Incidental Lung Nodule Dataset

The incidental lung nodules on CT scans are usually small nodules ($<$ 5mm) detected incidentally on cross-sectional imaging studies performed for some other reason. Unlike low-dose CT lung cancer screening suggested by multiple trials and studies [2], [53], [54], this dataset is collected from either contrast or non-contrast enhanced CT scans with normal dose at the Houston Methodist Hospital System. This can make incidental lung nodules difficult to detect; the patient may have other abnormalities present in the scan that can mislead the radiologist or classifier. Therefore, detecting them at the early stage and monitoring their growth over time can potentially help prevent them from developing into lung cancer in the future. Current lung nodule detection networks are typically trained on public low-dose CT datasets such as LUNA16 or NLST [16], [55]. These models have shown

TABLE I

PREDICTION PERFORMANCE OF DEEP NETWORKS TRAINED ON DATA-I AND TESTED ON DATA-I, DATA-II, AND DATA-III. WE PERFORMED 5-FOLD CROSS-VALIDATION ON THE SOURCE DATA (DATA-I) AND REPORT THE AVERAGE (±STD.) FOR EACH METRIC AND DATASET OVER THE MODELS.

Model	#param. (M)	Validation on different data (Data-I / Data-II / Data-III)			
		precision	recall	accuracy	AUROC
ResNet-20	0.8	90.7(±1.6) / 55.1(±1.8) / 9.5(±3.4)	59.7(±3.3) / 74.1(±2.1) / 80.9(±2.3)	86.7(±1.1) / 56.5(±2.4) / 37.4(±2.4)	78.6(±1.8) / 59.3(±2.7) / 57.3(±2.2)
ResNet-110	5.2	96.3(±1.4) / 58.1(±2.2) / 10.7(±2.8)	66.4(±4.3) / 86.5(±1.9) / 73.6(±2.1)	90.6(±2.8) / 61.9(±2.0) / 49.2(±1.8)	82.7(±2.3) / 67.7(±2.6) / 60.4(±2.0)
ResNet-1202	58.1	91.5(±1.7) / 72.1(±1.4) / 12.7(±2.1)	85.9(±2.7) / 64.5(±3.1) / 44.4(±4.0)	93.3(±1.4) / 69.6(±2.5) / 71.6(±2.0)	91.3(±1.8) / 69.7(±2.4) / 59.2(±2.3)
FastCapsNet	52.2	82.5(±2.0) / 63.5(±3.1) / 11.2 (±3.2)	88.2(±2.9) / 90.6(±0.7) / 74.5(±1.8)	91.9(±1.7) / 69.1(±2.7) / 51.2(±2.3)	88.6(±1.8) / 69.0(±2.3) / 61.8(±1.9)
FastCaps++	2.4	96.2(±1.5) / 75.8(±2.1) / 11.6(±2.6)	87.0(±3.1) / 66.9(±2.6) / 69.3(±2.4)	95.3(±2.0) / 72.3(±2.5) / 55.8(±2.1)	92.8(±1.6) / 72.3(±2.4) / 62.0(±2.1)

inferior performance in identifying incidental lung nodules from normal dose CT scans. On the other hand, the size of the incidental lung nodules is comparatively smaller which can be considered another factor causing a shift in the data distribution for the detection problem.

Due to the unique nature of incidental lung nodules and the complexity involved in detecting and annotating them, this dataset is relatively small compared to the other two datasets, and consists of 94 CT scans; 55 of which have at least one nodule. The original CT slices are of size 512×512 pixels for all patients with varying numbers of slices for different patients. A resampling step is conducted to achieve an isotropic resolution of $1 \times 1 \times 1$ mm so that the networks do not need to learn zoom/slice thickness invariance. Normalization is also used to compress the HU values to be in the range of 0 to 255. Positive samples are created by cropping a volume centered on the nodule. Negative samples are created by collecting the false-positive nodules generated by a 3D Faster R-CNN network [56] pre-trained on LUNA-16.

V. EXPERIMENTS AND RESULTS

A. Training Procedure and Implementations Details

All baseline deep models were trained using an Adam optimizer with $\beta_1 = 0.9$ ($\beta_1 = 0.5$ for CapsNets), $\beta_2 = 0.999$, a fixed batch size of 16, and a learning rate of 10^{-3} which was decayed exponentially (every 1000 steps with a base of 0.97) to a minimum of 10^{-5}. CNNs were trained with cross-entropy loss while CapsNets were trained to minimize the margin loss [43] to enforce the output capsule to have a large activation a if and only if a nodule exists in the input image:

$$\mathcal{L}_{\text{class}} = y \max(0, m^+ - a)^2 + \lambda(1 - y) \max(0, a - m^-)^2] \tag{7}$$

where y is the ground truth label. Minimizing this loss forces a to be higher than m^+ if a nodule exists, and lower than m^- otherwise. In our experiments, we set $m^+ = 0.9$, $m^- = 0.1$, and $\lambda = 0.5$. The parameters of the recurrent task network are optimized through maximizing the cross-entropy between predicted probabilistic scores and the ground truth labels. We train the MEMCAP model end-to-end using the ADAM optimizer with the same configuration as the baseline models. A random search is performed to find the best parameter values. The best results are achieved using 128 memory slots of size 40 and an LSTM controller with 200 hidden units.

For each episode, we randomly applied a combination of contrast, brightness, Gaussian blurring and affine transformations to the entire 3D space to generate samples from the new simulated domain (or task). Note that for fair comparison, the same data augmentation techniques is used in all of our experiments. The probability of each transformation being selected for a given simulated domain is 0.5. We used the Python Imaging Library (PIL) [57] to perform contrast, brightness and blurring transformations; parameters were selected in the range $[0.1, 1.9]$ to allow for both low and high brightness and contrast values (where 1 outputs the original image) and a range of $[0, 2]$ for blurring. We used TensorFlow [58] to apply affine transformations, including scaling ($[0.5, 2]$, where 0.5 corresponds to half the image size in each dimension), translation (up to 20 pixels along each axis), and shear (with counterclockwise shear angle selected in the range $[-0.5, 0.5]$ radians). All analyses are done using a desktop machine equipped with 128 GB of memory, a single NVidia TITAN V with 12 GB of video memory, and Intel Core i9-7960X CPU 2.80GHz with 32 cores.

B. Baseline Deep Neural Network Performance

To understand the effects of domain shifts between training and testing data, we trained various 3D models on DATA-I (i.e. LUNA) and then evaluated performance on all three datasets. We observe significant drops in accuracy across various popular computer vision architectures when testing on data from a different domain (as shown in Table I). This result was expected as these architectures are not explicitly designed to deal with the domain shift problem. This emphasizes the challenge of designing a framework that not only performs well on a provided set of samples, but also quickly adapts to a new domain by providing only a few samples without the need to retrain the whole system from scratch; especially in the medical imaging field where the annotation process is time-consuming, costly, and inconsistent.

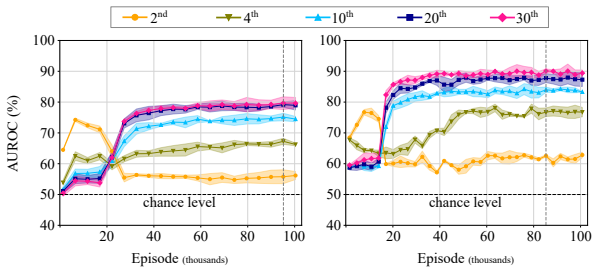

Fig. 5. Comparison of the test prediction performance of the proposed framework with (right) and without (left) the external memory on the LUNA dataset. The average AUROC (±std.) is computed over 100 unique meta-test tasks and after different numbers of labeled samples.

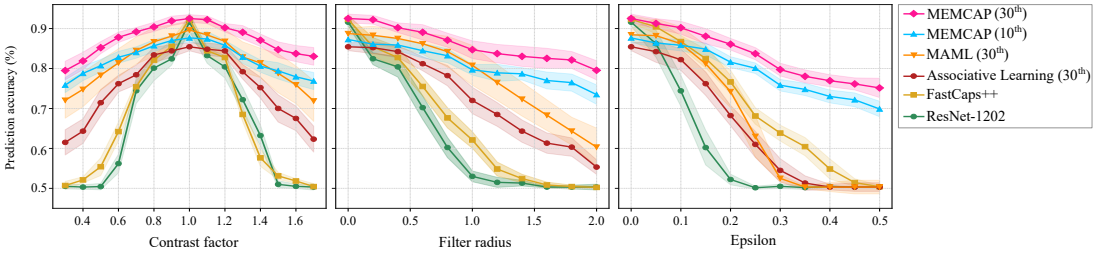

Fig. 6. Quantitative comparison of the test classification performance of various models in response to applying different types of distortions to the input images; namely, contrast shift (left), Gaussian blurring noise (middle), and FGSM adversarial attack (right). The shaded areas depict the standard deviation of prediction AUROC over 100 randomly sampled episodes.

TABLE II

COMPARING THE AVERAGE TEST AUROC (%) ACHIEVED BY DIFFERENT APPROACHES OVER 100 RUNS WITH RANDOMLY SELECTED ANNOTATED SAMPLES FROM THE TARGET DATASET. THE NUMBER OF PROVIDED LABELED SAMPLES FOR EACH EXPERIMENT ARE DENOTED ABOVE THE RESULTS (I.E. $\mathbf{2}^{\text{ND}}$ FOR TWO SAMPLES, $\mathbf{4}^{\text{TH}}$ FOR FOUR, AND SO ON).

Model	AUROC with Labeled Samples (%)														
	Data-I					Data-II					Data-III				
	2nd	4th	10th	20th	30th	2nd	4th	10th	20nd	30th	2nd	4th	10th	20th	30th
Transfer Learning	-	-	-	-	-	**70.16**	70.23	71.31	73.20	73.56	**62.04**	62.12	62.83	63.24	63.58
Associative Learning	**67.29**	72.33	77.81	80.07	80.45	65.58	**73.01**	75.16	78.94	80.19	**62.10**	**64.21**	68.49	70.63	72.06
LSTM	56.45	68.86	73.22	79.54	80.33	60.33	67.01	75.40	81.00	81.31	59.42	61.33	67.81	72.91	74.68
MANN	59.30	69.94	75.63	84.11	85.35	61.45	69.73	72.59	82.49	83.03	59.95	63.71	80.16	83.28	84.19
MAML	60.43	68.43	76.39	83.21	85.13	60.11	71.57	78.42	83.84	84.29	60.79	63.43	78.58	81.94	82.52
MEMCAP (ours)	62.77	**77.43**	**84.09**	**87.92**	**90.24**	64.17	**73.07**	**83.42**	**89.57**	**92.45**	61.23	**64.43**	**84.71**	**88.67**	**89.09**

C. Evaluation of adaptive classifier

To evaluate the domain adaptability of our memory-augmented architecture, we utilized an episodic training scheme with simulated domain shifts on Data-I. Note that the training and testing data subsets were disjointed to prevent data leakage. Fig. 5 shows a comparison of test performance with and without the external memory bank during the meta-testing phase as a function of the number of training episodes. Each training episode consists of randomly sampling a few images from both classes and applying the simulated domain shift on all of them. We evaluated the adaptation performance of the model after every few episodes, during which no further learning occurred and the network predicts the class labels for samples pulled from a disjoint set. We drew labeled samples (from 2 to 30 samples) from the test data and evaluated the classifiers on the remaining data. The labeled samples effectively mimic the data that needs to be annotated by the human expert when adapting to the new domain. For example, the 2^{nd} sample accuracy is the classification accuracy after providing two labeled samples from a given domain, while the 4^{th} sample accuracy is the classification accuracy after the first four labeled samples, and so on. We generate 100 unique meta-test tasks, each meant to mimic unique sample domains. We note that storing a single sample per class (yellow curve) allows the architecture to encode some relevant information which helps it exceed chance performance. The MEMCAP architecture achieves AUROC of 84.1% and 90.2%, after providing 10 and 30 labeled samples respectively. We also observe that one training episode of MEMCAP lasts only $0.47(\pm 0.04)$ seconds on average, with an overall training time of around 14 hours. The architecture is also composed of 3.6M trainable parameters (compared to MAML's 7.1M), requires 11.48 GB of memory (compared to MAML's 11.89 GB), and has a computational overhead of 178M FLOPs (compared to MAML's 195M).

We then evaluated the proposed architecture's robustness to various domain shifts. To do this, we took the best model (MEMCAP trained for 85,000 episodes) and tested it in two different experiments. The first consisted of evaluating the model with three different simulated domain shifts (Contrast shifts, Gaussian blurring, and FGSM Adversarial attacks [59]) of various magnitudes. FGSM Adversarial attacks compute the gradient of the loss with respect to each pixel and then alter pixel intensity in the opposite direction of the gradient which is scaled by a single hyperparameter ϵ. We also compare with the results obtained from learning by association [60] and MAML [38]. We modify the baseline models to make the comparison fair. For associative learning, we train the network on the source domain (Data-I) to minimize the loss function (a combination of visit, walker and classification loss as explained in [60]). For MAML, the best performance was achieved when setting the inner and outer update learning rates as $\alpha = 10^{-3}$ and $\beta = 0.005$. The results presented in Fig. 6 demonstrates that the MEMCAP model trained with 30 labeled samples per domain significantly outperforms other methods after both mild and severe data distortions are applied.

The second experiment was conducted by simply evaluating the performance of the model on CT scan datasets derived from different institutions. This enables us to check the networks' adaptability and performance robustness in a more natural setting. Classification results of the different models are presented in Table II. We also compare our results with those achieved using transfer learning in which the models trained on Data-I are fine-tuned to each respective target domain (i.e Data II and III) [61], [62]. After training the model on Data-I, we freeze the parameters of the convolutional layers and

only fine-tune the parameters of the capsule layers to prevent over-fitting on the small number of samples from Data-II and III. The initial learning rate is decreased to 10^{-4} and is decayed exponentially (every 1000 steps with a base of 0.97) to a minimum of 10^{-5}. We also performed the transfer learning experiment using all available labelled data from each target domain to determine the maximum performance that can be achieved with the baseline model. FastCaps++ achieves 95.2% and 92.1% AUROC on dataset II and III respectively. Considering the performance of MEMCAP as shown in Table II, we see that MEMCAP is able to achieve 97.1% and 96.8% of the theoretical maximum performance by using only 30 labeled samples from the respective target domains.

For the domain adaptation with associative learning, we used the network that was pre-trained under the associative learning regime and fine-tune it using a different number of labeled data from the target domain. Also note that a subset of 100 samples from the remaining training data was randomly selected as the set of unlabeled samples required for this approach. For all three baseline approaches, we ran the fine-tuning for as many iterations as required and reported the best results achieved. We observed that MEMCAP significantly outperforms the other baseline methods, with larger labeled sample sets contributing to more substantial increases in performance. We also generated results using the MANN architecture (with ResNet-110 as the feature extractor) to highlight the contribution of the FastCaps++ feature extractor [39]. Our result indicates that FastCaps++ in MEMCAP improves generalization with fewer samples in all tasks.

VI. Discussion

The adaptability of a CAD model to various different domains is fundamental to its feasibility in healthcare applications. This is due to the significant variation that can be seen across medical datasets; differences between patients, patient cohorts, and imaging centers can all contribute to potentially degraded CAD performance. In the context of Computed Tomography (CT) scans, we note that these variations are often caused by the use of unique CT scanners with varying HU values [17]. The reason for scanning a patient can also contribute to these significant domain shifts; the dosage of radiation used for each disease can vary, which alters the image noise distributions. Even data derived from various different imaging centers (as is the case for the LUNA dataset) can not characterize all of the observable domain variations. This is reflected in our results presented in Table I which show that deep neural networks trained on Data I do not generalize well to Data II, a similar sample set which was collected in a different institution. It is unrealistic to have physicians label data for each possible domain; even with a model trained on a large dataset, fine-tuning and retraining is required to match the new target domain. The use of domain adaptation methods is therefore warranted in the context of medical imaging tasks.

We propose MEMCAP, a deep neural network architecture trained with meta-learning to perform domain adaptation in lung nodule classification from CT scans. It consists of a CapsNet feature network that extracts invariant low and high-level semantic structures across domains from the high-dimensional input volume. The output is then fed to the task network: a memory augmented recurrent network which learns to quickly store and retrieve domain-specific information from its external memory bank using a small number of labeled samples. MEMCAP is thus able to leverage the available labeled target examples to store and exploit the underlying intrinsic information in the target domain.

To evaluate the performance of our proposed architecture, we compared our results with transfer learning, associative learning, and MAML. We simulated domain shifts by applying different types of distortions (contrast shift, Gaussian blurring, and FGSM adversarial attacks). From these results, we observe that MEMCAP was more robust to data distortions than the other evaluated methods. This suggests that MEMCAP was more resistant to domain shifts and was thus able to learn the underlying domain-independent information. We then evaluated the performance of the technique with data sets collected from unique institutions and with unique imaging settings. In Data II, the samples were collected from a single hospital and were then labeled by an independent group of experienced thoracic radiologists. We note that MEMCAP significantly outperforms the other techniques once enough labeled samples are provided; with just 10 labeled samples, MEMCAP achieves an 83.42% average AUROC, while MAML requires 20 labeled samples to achieve a similar level of performance. The performance disparity between MAML and MEMCAP is due to the inherent weaknesses of the MAML framework; while we observed stable performance across episodes in the MEMCAP model (as shown in the right panel of Figure 5), we observed significant performance instability when training the MAML model. This can be the result of vanishing and exploding gradients that occur due to a lack of skip connections and the large depth of the unfolded network [63]. We also note that the performance of MAML depends heavily on finding the optimal hyper-parameters across a large range of possible values (such as the α and β learning rates) for a given input dataset. MEMCAP is less susceptible to these limitations; it achieves its peak performance of 92.45% with 30 labeled samples, exceeding the performance of all baseline methods by a large margin.

We then evaluated the effectiveness of MEMCAP on the common occurrence of incidental lung nodules which are usually missed by radiologists and classifiers [64], [65]. These are lung nodules that were incidentally detected while scanning for another disease; this means that either contrast or non-contrast enhanced scans could be used to image patients which generally have a low-risk of lung cancer. This makes the detection of incidental lung nodules challenging; while screening for lung nodules in a high-risk population primes the radiologist for the task at hand, the assessment of incidental lung nodules must be done simultaneously to detect the other intended abnormalities. This issue is exacerbated by the fact that incidental lung nodules are detected in both non-contrast and contrast enhanced scans with imaging protocols optimized for different organs other than the lung. This is especially limiting as a thin slice thickness is usually required to identify a

small lung nodule. Unlike Data I and II, all samples generated in Data III were incidental. Moreover, due to the nature of these scans, the nodules are often smaller than the nodules observed in Data I and II. The results show that this significant domain shift is handled well by the MEMCAP framework when compared to the other baseline methods. We note that MEMCAP outperforms all other methods and achieves 84.7% and 89.1% accuracy for classifying incidental lung nodules after only 10 and 30 labeled samples, respectively. Given the resistance of MEMCAPS to domain shifts, we show that it is well suited to the task of assessing incidental lung nodules, and can be used effectively in a clinical setting.

We can also conclude based on our findings that this architecture is well suited to other medical imaging techniques. For example, optical coherence tomography (OCT) is a medical imaging technique that utilizes low-coherence light to capture both 2D and 3D structures within biological tissues at both low and high spatial resolution [66]. Previous works have shown that it is possible to augment the sensitivity and specificity of OCT skin lesion analysis through the integration of computational techniques [67]. Future works should aim to integrate MEMCAP into OCT skin lesion analysis in order to address both the data scarcity and resolution variability of the data.

VII. Conclusion

This paper systematically evaluates the adaptability of deep networks; we found that while deep neural networks achieve state-of-the-art performance on a set of data, they do not perform well in response to domain shifts. We propose a practical adaptive classifier called MEMCAP which is capable of taking a few annotated inputs from a new target domain to refine its decision making ability accordingly. This prevents the operator from having to re-train the network for each domain, as re-training requires a significant amount of time, computational resources, and human effort. Our experimental results have demonstrated that when the data distribution changes, the proposed classifier adapts almost perfectly in the lung nodule classification task while popular deep networks' performance decrease to chance with large domain shifts. Exploring the possibility of utilizing the proposed framework for the recognition of critical radiology findings from across the body (such as liver tumors, enlarged lymph nodes, and so on) as well as examining different optimization strategies to speed up the convergence of MEMCAP are promising directions for future work.

References

[1] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2017," *CA: a cancer journal for clinicians*, vol. 67, no. 1, pp. 7–30, 2017.

[2] N. L. S. T. R. Team *et al.*, "Reduced lung-cancer mortality with low-dose computed tomographic screening," *N Engl J Med*, vol. 2011, no. 365, pp. 395–409, 2011.

[3] A. Brady, R. Ó. Laoide, P. McCarthy, and R. McDermott, "Discrepancy and error in radiology: concepts, causes and consequences," *The Ulster medical journal*, vol. 81, no. 1, p. 3, 2012.

[4] A. P. Brady, "Error and discrepancy in radiology: inevitable or avoidable?" *Insights into imaging*, pp. 1–12, 2016.

[5] T. N. Shewaye and A. A. Mekonnen, "Benign-malignant lung nodule classification with geometric and appearance histogram features," *arXiv preprint arXiv:1605.08350*, 2016.

[6] K. Awai, K. Murao, A. Ozawa, M. Komi, H. Hayakawa, S. Hori, and Y. Nishimura, "Pulmonary nodules at chest ct: effect of computer-aided diagnosis on radiologists' detection performance," *Radiology*, vol. 230, no. 2, pp. 347–352, 2004.

[7] Q. Dou, H. Chen, L. Yu, J. Qin, and P.-A. Heng, "Multilevel contextual 3-d cnns for false positive reduction in pulmonary nodule detection," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 7, pp. 1558–1567, 2017.

[8] M. Firmino, A. H. Morais, R. M. Mendoça, M. R. Dantas, H. R. Hekis, and R. Valentim, "Computer-aided detection system for lung cancer in computed tomography scans: review and future prospects," *Biomedical engineering online*, vol. 13, no. 1, p. 41, 2014.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[11] Y. Xu, T. Mo, Q. Feng, P. Zhong, M. Lai, I. Eric, and C. Chang, "Deep learning of feature representation with multiple instance learning for medical image analysis," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1626–1630.

[12] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016.

[13] Y. Bar, I. Diamant, L. Wolf, S. Lieberman, E. Konen, and H. Greenspan, "Chest pathology detection using deep learning with non-medical training," in *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*. IEEE, 2015, pp. 294–297.

[14] A. Mobiny and H. Van Nguyen, "Fast capsnet for lung cancer screening," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 741–749.

[15] Q. Wang, F. Milletari, H. V. Nguyen, S. Albarqouni, M. J. Cardoso, N. Rieke, Z. Xu, K. Kamnitsas, V. Patel, B. Roysam *et al.*, *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data: First MICCAI Workshop, DART 2019, and First International Workshop, MIL3ID 2019, Shenzhen, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13 and 17, 2019, Proceedings*. Springer Nature, 2019, vol. 11795.

[16] A. A. A. Setio, A. Traverso, T. De Bel, M. S. Berens, C. van den Bogaard, P. Cerello, H. Chen, Q. Dou, M. E. Fantacci, B. Geurts *et al.*, "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge," *Medical image analysis*, vol. 42, pp. 1–13, 2017.

[17] A. M. A. Roa, H. K. Andersen, and A. C. T. Martinsen, "Ct image quality over time: comparison of image quality for six different ct scanners over a six-year period," *Journal of applied clinical medical physics*, vol. 16, no. 2, pp. 350–365, 2015.

[18] A. Kumar, A. Saha, and H. Daume, "Co-regularization based semi-supervised domain adaptation," in *Advances in neural information processing systems*, 2010, pp. 478–486.

[19] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," *arXiv preprint arXiv:1711.03213*, 2017.

[20] P. Monkam, S. Qi, H. Ma, W. Gao, Y. Yao, and W. Qian, "Detection and classification of pulmonary nodules using convolutional neural networks: A survey," *IEEE Access*, vol. 7, pp. 78 075–78 091, 2019.

[21] J. Zhang, Y. Xia, H. Cui, and Y. Zhang, "Pulmonary nodule detection in medical images: A survey," *Biomedical Signal Processing and Control*, vol. 43, pp. 138–147, 2018.

[22] H. Daumé III, "Frustratingly easy domain adaptation," *arXiv preprint arXiv:0907.1815*, 2009.

[23] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *arXiv preprint arXiv:1409.7495*, 2014.

[24] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 361–368.

[25] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," *ECCV 2010*, pp. 213–226, 2010.

[26] H. Van Nguyen, K. Zhou, and R. Vemulapalli, "Cross-domain synthesis of medical images using efficient location-sensitive deep network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 677–684.

[27] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647–655.

[28] H. V. Nguyen, H. T. Ho, V. M. Patel, and R. Chellappa, "DASH-N: Joint hierarchical domain adaptation and feature learning," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5479–5491, 2015.

[29] A. A. Rusu, M. Vecerik, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," *arXiv preprint arXiv:1610.04286*, 2016.

[30] J. Hoffman, E. Tzeng, J. Donahue, Y. Jia, K. Saenko, and T. Darrell, "One-shot adaptation of supervised deep convolutional models," *arXiv preprint arXiv:1312.6204*, 2013.

[31] A. Bergamo and L. Torresani, "Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach," in *Advances in neural information processing systems*, 2010, pp. 181–189.

[32] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2252–2259.

[33] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*. Springer, 2010, pp. 213–226.

[34] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3723–3732.

[35] G. Marcus, "Deep learning: A critical appraisal," *arXiv preprint arXiv:1801.00631*, 2018.

[36] J. Schmidhuber, "Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook," Ph.D. dissertation, Technische Universität München, 1987.

[37] S. Thrun and L. Pratt, "Learning to learn: Introduction and overview," in *Learning to learn*. Springer, 1998, pp. 3–17.

[38] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1126–1135.

[39] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks," *arXiv preprint arXiv:1605.06065*, 2016.

[40] C. I. Bercea, O. Pauly, A. Maier, and F. C. Ghesu, "Shamann: shared memory augmented neural networks," in *International Conference on Information Processing in Medical Imaging*. Springer, 2019, pp. 830–841.

[41] C. Zhang, Y. Wang, X. Zhao, Y. Guo, G. Xie, C. Lv, and B. Lv, "Memory-augmented anomaly generative adversarial network for retinal oct images screening," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 1971–1974.

[42] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[43] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, 2017, pp. 3856–3866.

[44] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with em routing," in *International conference on learning representations*, 2018.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[46] A. Mobiny, H. Lu, H. V. Nguyen, B. Roysam, and N. Varadarajan, "Automated classification of apoptosis in phase contrast microscopy using capsule network," *IEEE transactions on medical imaging*, vol. 39, no. 1, pp. 1–10, 2019.

[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[48] C. Giraud-Carrier, R. Vilalta, and P. Brazdil, "Introduction to the special issue on meta-learning," *Machine learning*, vol. 54, no. 3, pp. 187–193, 2004.

[49] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[50] Q. Dou, C. de Castro, K. Kamnitsas, and B. Glocker, "Domain generalization via model-agnostic learning of semantic features," in *Advances in Neural Information Processing Systems*, 2019, pp. 6447–6458.

[51] M. Woodward and C. Finn, "Active one-shot learning," *arXiv preprint arXiv:1702.06559*, 2017.

[52] S. G. Armato, G. McLennan, L. Bidaut, M. F. McNitt-Gray, C. R. Meyer, A. P. Reeves, B. Zhao, D. R. Aberle, C. I. Henschke, E. A. Hoffman *et al.*, "The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans," *Medical physics*, vol. 38, no. 2, pp. 915–931, 2011.

[53] A. Jemal and S. A. Fedewa, "Lung cancer screening with low-dose computed tomography in the united states—2010 to 2015," *JAMA oncology*, vol. 3, no. 9, pp. 1278–1281, 2017.

[54] W. C. Black, I. F. Gareen, S. S. Soneji, J. D. Sicks, E. B. Keeler, D. R. Aberle, A. Naeim, T. R. Church, G. A. Silvestri, J. Gorelick *et al.*, "Cost-effectiveness of ct screening in the national lung screening trial," *N Engl J Med*, vol. 371, pp. 1793–1802, 2014.

[55] N. L. S. T. R. Team, "The national lung screening trial: overview and study design," *Radiology*, vol. 258, no. 1, pp. 243–253, 2011.

[56] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[57] F. Lundh, M. Ellis *et al.*, "Python imaging library (pil)," 2012.

[58] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.

[59] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[60] P. Haeusser, A. Mordvintsev, and D. Cremers, "Learning by association-a versatile semi-supervised training method for neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 3, no. 5, 2017, p. 6.

[61] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[62] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[63] A. Antoniou, H. Edwards, and A. Storkey, "How to train your maml," *arXiv preprint arXiv:1810.09502*, 2018.

[64] R. Hossain, C. C. Wu, P. M. de Groot, B. W. Carter, M. D. Gilman, and G. F. Abbott, "Missed lung cancer," *Radiologic Clinics*, vol. 56, no. 3, pp. 365–375, 2018.

[65] H. MacMahon, J. H. Austin, G. Gamsu, C. J. Herold, J. R. Jett, D. P. Naidich, E. F. Patz Jr, and S. J. Swensen, "Guidelines for management of small pulmonary nodules detected on ct scans: a statement from the fleischner society," *Radiology*, vol. 237, no. 2, pp. 395–400, 2005.

[66] S. O'Leary, A. Fotouhi, D. Turk, P. Sriranga, A. Rajabi-Estarabadi, K. Nouri, S. Daveluy, D. Mehregan, and M. Nasiriavanaki, "Oct image atlas of healthy skin on sun-exposed areas," *Skin Research and Technology*, vol. 24, no. 4, pp. 570–586, 2018.

[67] Z. Turani, E. Fatemizadeh, T. Blumetti, S. Daveluy, A. F. Moraes, W. Chen, D. Mehregan, P. E. Andersen, and M. Nasiriavanaki, "Optical radiomic signatures derived from optical coherence tomography images improve identification of melanoma," *Cancer Research*, vol. 79, no. 8, pp. 2021–2030, 2019. [Online]. Available: https://cancerres.aacrjournals.org/content/79/8/2021