Few-Shot Learning via Repurposing Ensemble of Black-Box Models

Minh Hoang[1], **Trong Nghia Hoang**[2]

[1] Lewis-Sigler Institute of Integrative Genomics, Princeton University, Princeton, NJ 08540
[2]School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99613
minhhoang@princeton.edu, trongnghia.hoang@wsu.edu

Abstract

This paper investigates the problem of exploiting existing solution models of previous tasks to address a related target task with limited training data. Existing approaches addressing this problem often require access to the internal parameterization of the existing solution models and possibly their training data, which is not possible in many practical settings. To relax this requirement, We approach this problem from a new perspective of black-box re-purposing, which augments the target inputs and leverages their corresponding outputs generated by existing black-box APIs into a feature ensemble. We hypothesize that such feature ensemble can be learned to incorporate and encode relevant black-box knowledge into the feature representation of target data, which will compensate for their scarcity. This hypothesis is confirmed via the reported successes of our proposed black-box ensemble in solving multiple few-shot learning tasks derived from various benchmark datasets. All reported results show consistently that the set of heterogeneous black-box solutions of previous tasks can indeed be reused and combined effectively to solve a reasonably related target task without requiring access to a large training dataset. This is the first step towards enabling new possibilities to further supplement existing techniques in transfer or meta learning with black-box knowledge.

1 Introduction

Learning in few-shot settings generally requires (1) distilling transferable knowledge from existing solution models, which were previously trained to solve other related tasks; and then (2) recomposing them appropriately in new contexts. This is in fact a common recipe that was seen across numerous existing solution paradigms to few-shot learning, which include transfer learning (Du et al. 2020; Pan and Yang 2009; Tripuraneni, Jordan, and Jin 2020), multi-task learning (Ben-David and Schuller 2003; Bonilla, Chai, and Williams 2008; Fifty et al. 2021) and meta learning (Dinh, Nguyen, and Nguyen 2020; Fallah, Mokhtari, and Ozdaglar 2020; Finn, Abbeel, and Levine 2017; Yoon et al. 2018).

The underlied principle here is that well-trained models on related tasks can be exploited to compensate for the lack of data on a target task. This can be enabled via either fine-tuning a modifiable pre-trained model with target domain data (transfer learning); synchronizing the training processes across different tasks to transform multiple sources of local, heterogeneous data into a unified, richer source of learning feedback for all models (multi-task learning); or distilling the common inferential knowledge of the resulting models into a base model that can solve any unseen tasks (meta learning). Alternatively, there also exists a more recent line of data-free meta learning approaches (Lam et al. 2021; Wang et al. 2022) that predict the model parameters given the task identifier's representation, which requires access to the internal parameterization of pre-trained solution models of previous tasks, but not their training data. However, most approaches in these directions require access to the architectures and internal parameterizations of existing models, or even the labeled data and training processes that were used to generate them. Hence, they are not applicable to modern practices where pre-trained models are often released as black-box functions. For example, these include the Machine Learning-as-a-Service (MLaaS) toolsets provided by Microsoft Custom Vision and Amazon SageMaker.

To mitigate this limitation, another line of research on black-box model fusion (Hoang et al. 2020) or reuse (Hoang et al. 2019a,b; Wu, Liu, and Zhou 2019; Shao et al. 2021; Wu et al. 2021) can be considered as a viable alternative. In particular, Hoang et al. (2020) adopts a generic two-step approach which first collects and embeds the input-output responses of these frozen black-box models into a latent space that factorizes into task-agnostic and task-specific inferential patterns. Given a new task, the distilled task-agnostic patterns can then be put together into an implicitly represented base model, which can be fine-tuned with domain data to produce a customized model. Nonetheless, this approach assumes that the optimal solution model exists on a relatively simple weight space, e.g. feed-forward neural networks, and can be easily recomposed from the distilled patterns. This assumption limits their effectiveness to low-complexity tasks such as simple regression or MNIST classification (LeCun, Cortes, and Burges 2010), and precludes extension to domains that requires more sophisticated architectures such as convolutional neural networks. As another alternative, the model reuse literature (Wu, Liu, and Zhou 2019; Shao et al. 2021; Wu et al. 2021; Hoang et al. 2019b,a) instead assumes a black-box sharing protocol where extra statistics summarizing the private data are available; or that

pre-trained models share the same output space, which also limits the applicability to more practical scenarios. To mitigate such limitation, this paper explores a complementary approach where instead of dissecting the black boxes into transferable patterns, the focus is on augmenting the target input and leveraging their corresponding black-box outputs into an enriched feature representation to compensate for the lack of training data. In a single-source transfer task, the previous work of (Tsai, Chen, and Ho 2020) has shown that learning such augmentation can help unlock black-box knowledge within a complex architecture (e.g., ResNet (He et al. 2015)) pre-trained with a large amount of generic data (e.g., ImageNet (Deng et al. 2009)) to solve a specific task with limited training data.

Motivated by this success, we generalize this perspective to re-purpose and leverage the black-box knowledge within multiple existing solutions of different tasks, which were trained on diverse specialized data rather than a common generic data sources, into a more holistic model to solve a related task with limited data. Unlike the single-model setup in (Tsai, Chen, and Ho 2020), which assumes that the black-box model has already internalized sufficient generic data to solve the target task, we consider settings where each pre-trained model alone does not have enough information to solve the new task. Hence, combining them is necessary and can be achieved via the following contributions:

1. We develop a black-box re-purposing framework that maximizes a statistical co-occurrence between the target label and an ensemble of black-box output on a (learnable) input augmentation. This is parameterized via a set of flow-based transformations (Dinh, Krueger, and Bengio 2015) that augment the target input to extract the corresponding output from each pre-trained black box (Section 3.1).

2. We design specific parameterization for the above re-purposing framework, expressed in terms of its input augmentation and black-box output ensemble, which can be used as a medium to distill and fine-tune predictive knowledge from pre-trained task models. The entire distillation and fine-tuning process is guided by a few-shot dataset of a related task (Section 3.2).

3. We conduct experimental studies showcasing the effectiveness of the proposed methods on variety of realistic task domains with high-complexity task models (Section 4). For better clarity, we also provide a succinct review on existing, related bodies of literature in Section 2 below.

2 Related Work

Transfer Learning (Pan and Yang 2009) is a popular approach for learning in new environments with limited training data. Its key idea is to use existing models well-trained on related environments to learn a good feature extractor (Bengio, Courville, and Vincent 2013) for data in a target environment. For example, in natural language processing, pre-trained neural language models are often used to generate sentence embeddings (Qiu et al. 2020) in many downstream tasks. The induced representation captures the common structure across tasks (e.g., representation of words) and reduces the complexity of the hypothesis space (Ben-David and Schuller 2003). Theoretical justifications of this insight have been explored in the previous works of (Baxter 2000; Ben-David and Schuller 2003; Du et al. 2020) and (Tripuraneni, Jordan, and Jin 2020).

Meta Learning (Finn, Abbeel, and Levine 2017) is another class of techniques (Yoon et al. 2018; Fallah, Mokhtari, and Ozdaglar 2020; Dinh, Nguyen, and Nguyen 2020) that aim to exploit the relatedness of tasks to learn a common initial model that can be quickly adapted to unseen tasks. The meta learning setting does not assume access to a previously identified source task with rich data that can be trained and specialized (via fine-tuning) to solve a target task. Instead, it requires sampling access to a distribution of related tasks and their model architectures to coordinate the training process. This is, however, not possible when the solution models were independently pre-trained in isolation and are neither transparent nor modifiable, such as cloud-based machine learning models.

Black-Box Transfer Learning. Most application of transfer learning (Pan and Yang 2009) and meta learning (Finn, Abbeel, and Levine 2017) assume access to both source and target task's data ahead of time, as well as the source task's pre-trained model's learned parameterizations. This is not applicable to scenarios where the trained models are released as black-box functions. To sidestep this limitation, (Tsai, Chen, and Ho 2020) proposes to learn a pair of input transformation and output mapping functions that collectively reprogram the source model to solve any target task. However, the developed technique is grounded in settings where sufficient domain data are available to train the reprogrammer model. Our work instead investigates the more challenging few-shot model repurposing problem, for which the lack of domain data will necessitate more efficient exploitation of the black-box artifacts.

3 Multi-Task Model Repurposing

Using the change-of-variable trick for density functions (Kestelman 1961), we can re-parameterize the target task's input-output distribution in terms of any feature-output distribution. This will allow us to learn an input augmentation for each existing black-box API that extracts informative features for the target task (see Lemma 3.1). These augmentations are modeled using invertible normalizing flows to prevent information loss (Dinh, Krueger, and Bengio 2015; Dinh, Sohl-Dickstein, and Bengio 2017; Rezende and Mohamed 2016; Kingma and Dhariwal 2018), and can be learned to maximize the target task's training data likelihood. Intuitively, this helps learn feature extractors that unlock most relevant knowledge from the pre-trained black-box APIs, which can be integrated into the target input's feature representation to compensate for its lack of training data. An overview of our workflow is shown in Fig. 1.

3.1 Flow-Based Multi-Task Embedding Model

Let $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_n$ denote n black-box models that were previously trained to solve n related but different tasks $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$. Each task $\mathcal{P}_i$ can be considered as a distribution over a product space $\mathcal{X}_i \times \mathcal{C}_i$ of input-output pair

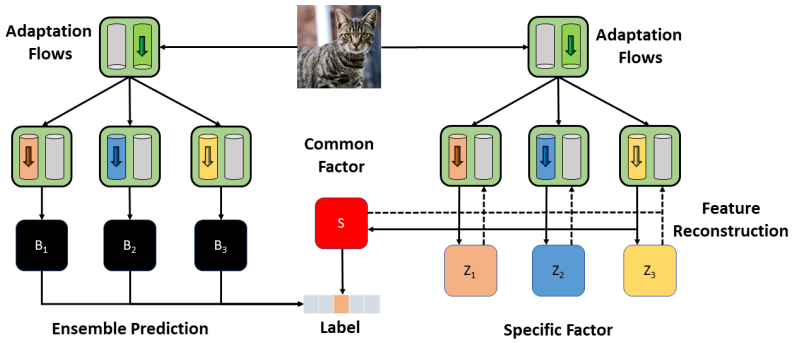


Figure 1: The workflow diagram of our model repurposing algorithm. The target image is passed through a set of adaptation flows to produce a set of augmented inputs. The augmented inputs are then decomposed into common and specific factors. The decomposition component is trained via an adapted VAE (Kingma and Welling 2013) framework that minimizes the feature reconstruction loss. The augmented inputs are also fed into a (learnable) black-box ensemble whose outputs are leveraged into an enriched feature representation for the target task, which is a part of the VAE loss, as described previously in Eq. (3.2).

$(\mathbf{x}_i, c_i)$. Here, we flexibly model c_i as the soft output of an oracle on $\mathbf{x}_i$ since the hard label y_i of $\mathbf{x}_i$ was never observed. Our goal is to extract and incorporate inferential knowledge from $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_n$ into a solution model $\mathbf{B}_*$ for a generalized task $\mathcal{P}_*$, given a limited amount of data. However, unlike white-box meta or multi-task learning, the models $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_n$ are not concurrently and synchronously trained to distill $\mathbf{B}_*$. Instead, those are pre-trained models whose weights are neither observable nor modifiable. In addition, these models are trained on different tasks with different input and output spaces. There is also no access to their training data. We will, however, show that it is still possible to distil their inferential knowledge via prompting them with appropriate input augmentation.

This is built upon Lemma 3.1, which characterizes the likelihood of the target task's data in terms of a likelihood of any invertible input augmentation. This can be further parameterized with the corresponding black-box APIs' output (see Section 3.2). As we learn the augmentation parameterization so that it maximizes the target data's likelihood (see Lemma 3.2), the latent knowledge within each black-box API can be distilled into its corresponding output to the augmented input, which can be incorporated into a feature ensemble that induces an effective solution to the target task.

Lemma 3.1. Let $p(\mathbf{x}_*, c_*)$ denote the density function of the target task's distribution $\mathcal{P}_*$ on an input-output pair $(\mathbf{x}_*, c_*)$. Let $h_i(\mathbf{x}_*)$ denote an invertible input augmentation function,

$$\log p\Big(\mathbf{x}_*, c_*\Big) \;=\; \log p\Big(h_i(\mathbf{x}_*), c_*\Big) + \log \left|\frac{\mathrm{d}h_i(\mathbf{x}_*)}{\mathrm{d}\mathbf{x}_*}\right| \quad (3.1)$$

where $h_i(\mathbf{x}_*)$ denote an input augmentation to extract knowledge from $\mathbf{B}_i$, which can be parameterized by any valid normalizing flow (Rezende and Mohamed 2016). This follows from the change-of-variable theorem (Appendix B).

Next, we parameterize a factorized embedding space for the augmented input $h_i(\mathbf{x}_*)$ whose latent coordinates comprise two orthogonal components $\mathbf{s}$ and $\mathbf{z}_i$. According to this factorization, $\mathbf{s}$ denotes a common latent factor across all tasks, whereas $\mathbf{z}_i$ characterizes a private latent component that is specific to task $\mathcal{P}_i$. To ensure this augmentation mechanism induces the most informative feature output (for the target task) from the black-box APIs, we need to further parameterize $p(h_i(\mathbf{x}_*), c_*)$ with the corresponding black-box output $\mathbf{B}_i(h_i(\mathbf{x}_*))$ and optimize the overall parameterization to maximize the RHS of Eq. (3.1). To ease the difficulty of a direct parameterization and optimization, we instead leverage a variational inequality in the previous work of (Kingma and Welling 2013) to derive a lower-bound for $\log p(h_i(\mathbf{x}_*), c_*)$, which is configurable via any suitable choices of a surrogate posterior $q(\mathbf{s}, \mathbf{z}_i \mid h_i(\mathbf{x}_*)) = q(\mathbf{s} \mid h_i(\mathbf{x}_*))q(\mathbf{z}_i \mid h_i(\mathbf{x}_*))$. This results in a more modular form for optimization as detailed in Lemma 3.2 below.

Lemma 3.2. Assuming the factorization $p(h_i(\mathbf{x}_*), c_*, \mathbf{s}, \mathbf{z}_i) = p(h_i(\mathbf{x}_*) \mid \mathbf{s}, \mathbf{z}_i) \cdot p(c_* \mid \mathbf{s}, h_i(\mathbf{x}_*)) \cdot p(\mathbf{s}, \mathbf{z}_i)$, we have $\log p(h_i(\mathbf{x}_*), c_*) \geq \mathrm{ELBO}(h_i(\mathbf{x}_*), c_*)$, where

$$\begin{aligned}\mathrm{ELBO}\Big(\,\cdot\,\Big) \;&\triangleq\; \mathbb{E}_{q(\mathbf{s},\mathbf{z}_i|h_i(\mathbf{x}_*))}\Big[\log p\Big(h_i(\mathbf{x}_*) \mid \mathbf{s}, \mathbf{z}_i\Big)\Big] \\ &+\; \mathbb{E}_{q(\mathbf{s}|h_i(\mathbf{x}_*))}\Big[\log p\Big(c_* \mid \mathbf{s}, h_i(\mathbf{x}_*)\Big)\Big] \\ &-\; \mathbb{D}_{\mathrm{KL}}\left(q\Big(\mathbf{s}, \mathbf{z}_i \mid h_i(\mathbf{x}_*)\Big) \| p\Big(\mathbf{s}, \mathbf{z}_i\Big)\right) \quad (3.2)\end{aligned}$$

Here, $p(c_* \mid \mathbf{s}, h_i(\mathbf{x}_*))$ can be further parameterized with the black-box output $\mathbf{B}_i(h_i(\mathbf{x}_*))$ (see Section 3.2). The detailed derivation of Eq. (3.2) is deferred to Appendix C.

Remark. The interested readers are referred to the extended version of this paper[1] for all appendices.

[1] https://htnghia87.github.io/publication/aaai2024a

Using Lemma 3.2, we can plug Eq. (3.2) into Eq. (3.1) to acquire $\log p(\mathbf{x}_*, c_*) \geq (1/n) \sum_{i=1}^{n} \mathbf{F}_i(\mathbf{x}_*, c_*)$, where

$$\mathbf{F}_i\Big(\mathbf{x}_*, c_*\Big) \triangleq \mathrm{ELBO}\Big(\,\cdot\,\Big) + \log \left| \frac{\mathrm{d}h_i(\mathbf{x}_*)}{\mathrm{d}\mathbf{x}_*} \right| . \quad (3.3)$$

Thus, we can sample $(\mathbf{x}_*, c_*) \sim \mathcal{P}_*$ and optimize for the parameterizations of the invertible function $h_i(\mathbf{x}_*)$, the surrogate posterior $q(\mathbf{s}, \mathbf{z}_i \mid h_i(\mathbf{x}_*))$ and the generative conditionals $p(c_*|\mathbf{s}, h_i(\mathbf{x}_*))$ and $p(h_i(\mathbf{x}_*) \mid \mathbf{s}, \mathbf{z}_*)$, such that the averaged value of $(1/n) \sum_{i=1}^{n} \mathbf{F}_i(\mathbf{x}_*, c_*)$ over the drawn samples $(\mathbf{x}_*, c_*)$ is maximized. Such sample set can be associated with the few-shot data that defines $\mathcal{P}_*$. Once the above is learned, the solution model for $\mathcal{P}_*$ is accessible via

$$\mathbf{B}_*(\mathbf{x}_*) \triangleq \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{q(\mathbf{s}|h_i(\mathbf{x}_*))} \left[\mathbb{E}_{p(c_*|\mathbf{s}, h_i(\mathbf{x}_*))} \Big[c_* \Big] \right] , \quad (3.4)$$

where we sample the prediction $y_* \sim \mathrm{Cat}(\mathrm{softmax}(c_*))$ with $c_* \triangleq \mathbf{B}_*(\mathbf{x}_*)$ being the pre-softmax output of $\mathbf{B}_*$.

3.2 Multi-Task Embedding Parameterization

Having defined the generic workflow to induce a solution model for the target task, we now complete our model specification by providing explicit parameterization for the aforementioned generative and input transformation components. Specifically, we will use the related task models $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_n$ to parameterize them, thereby drawing a (learnable) connection between their solution models and the distributional embedding patterns of the (target) task's data. This helps combine the solution patterns of these source tasks to synthesize a solution for the target task.

Input Transformation As described above, the input augmentation function $h_i(\mathbf{x}_*)$ is parameterized with a flow-based generative model which is composed of p sequential blocks. Each block comprises interleaving flows of different types, including $\mathrm{PlanarFlow}$, $\mathrm{RadialFlow}$ (Rezende and Mohamed 2016) and/or NVP (Dinh, Sohl-Dickstein, and Bengio 2017). The same flow architecture is used for all input augmentation functions $h_1(\mathbf{x}_*), h_2(\mathbf{x}_*), \ldots, h_n(\mathbf{x}_*)$ but their parameterized weights are not tied.

Generative Components There are three generative components $p(\mathbf{s}, \mathbf{z}_i)$, $p(c_*|\mathbf{s}, h_i(\mathbf{x}_*))$ and $p(h_i(\mathbf{x}_*)|\mathbf{s}, \mathbf{z}_i)$ in our multi-task embedding model, as specified in Section 3.1. In finer details, we factorize the prior $p(\mathbf{s}, \mathbf{z}_i) = p(\mathbf{s})p(\mathbf{z}_i)$ across the common and specific latent factors which underlies the input distribution of sub-task $\mathcal{P}_i$. Both priors are modeled as multivariate Gaussian with learnable mean and covariance matrix, which are initialized with $\mathbb{N}(\mathbf{0}, \mathbf{I})$ but are updated separately. The likelihood component $p(c_* \mid \mathbf{s}, h_i(\mathbf{x}_*))$ is modeled with a deep generative net that combines the predictions of the task models $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_n$ on the input transformation $h_i(\mathbf{x}_*)$. As the prediction outputs of the task models are on different output spaces, we further align them via a learnable mapping function $\mathbb{M}_\mathbf{s}$ induced by the common latent factor $\mathbf{s}$. The likelihood component is modelled as $p(c_*|\mathbf{s}, h_i(\mathbf{x}_*)) \triangleq c_*^\top w\left(\mathbf{s}, h_i(\mathbf{x}_*)\right)$, where

$$w(\mathbf{s}, h_i(\mathbf{x}_*)) \triangleq \sigma\left(\mathbb{M}_\mathbf{s}\left(\mathrm{Cat}\left[\mathbf{B}_a\Big(h_a(\mathbf{x}_*)\Big)\right]\right)\right) . \quad (3.5)$$

Here, σ denotes the softmax activation. The mapping function $\mathbb{M}_\mathbf{s}$ is parameterized as a composition of a feed-forward net with ReLU activation and a $\mathrm{Gumbel\text{-}softmax}$ layer that returns a discrete mapping from the concatenated output to the target output. The weight of its last layer is generated using a feed-forward net conditioned on $\mathbf{s}$.

Finally, the generative likelihood $p(h_i(\mathbf{x}_*) \mid \mathbf{s}, \mathbf{z}_i)$ is modeled as an isotropic Gaussian centered at $m(\mathrm{concat}[\mathbf{s}, \mathbf{z}_i])$ where m is a sequence of interleaving de-convolution, pooling and feed-forward net with intermediate ReLU and final Tanh activation functions. The specifics of these de-convolution, pooling and feed-forward nets depend on the application domain and are deferred to Appendix D.

Posterior Surrogates As described in Section 3.1, there are two posterior components, $q(\mathbf{s} \mid h_i(\mathbf{x}_*))$ and $q(\mathbf{z}_i \mid h_i(\mathbf{x}_*))$, which correspond to the common and specific latent coordinates of the target task's embedding space. Both are parameterized with multivariate Gaussian $\mathbb{N}(m(\mathbf{x}_*), \mathrm{diag}[v(\mathbf{x}_*)])$ where $m(\mathbf{x}_*)$ and $v(\mathbf{x}_*)$ are in turn ensembles of neural nets:

$$m(\mathbf{x}_*) = \sum_{i=1}^{n} w_i(\mathbf{x}_*) \cdot m_i(\mathbf{x}_*); v(\mathbf{x}_*) = \sum_{i=1}^{n} w_i(\mathbf{x}_*) \cdot v_i(\mathbf{x}_*)$$

where $w(\mathbf{x}_*) = [w_1(\mathbf{x}_*), \ldots, w_n(\mathbf{x}_*)]$ are generated by a learnable convolutional neural net. The neural nets $m_i(\mathbf{x}_*)$ and $v_i(\mathbf{x}_*)$ are parameterized as cascades of interleaving convolution, pooling and feed-forward layers, with intermediate ReLU and Tanh activation functions. The exact configurations of these layers depend on the specifics of the application domain and are deferred to Appendix D.

Random Gradient Estimation An important point to note from the above parameterization is that in the aforementioned, the parameterization of the likelihood $p(c_* \mid \mathbf{s}, h_i(\mathbf{x}_*))$ involves the black-box models, which cannot propagate gradients due to their immovable and inaccessible inner weights. As such, we need to estimate the gradient $\nabla_{\mathbf{x}_*} \mathbf{B}_i(\mathbf{x}_*)$ via finite difference methods. In particular, let $\mathbf{z} \sim \mathbb{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{v} = \mathbf{z}/\|\mathbf{z}\|$, it follows that $\mathbf{v}$ is a unit vector and $\nabla_{\mathbf{x}_*} \mathbf{B}_i(\mathbf{x}_*)^\top \mathbf{v} = \mathcal{D}_\mathbf{v} \mathbf{B}_i(\mathbf{x}_*)$ which is the directional gradient of $\mathbf{B}_i(\mathbf{x}_*)$. This is also defined as:

$$\mathcal{D}_\mathbf{v} \mathbf{B}_i(\mathbf{x}_*) \triangleq \lim_{\alpha \to 0} \frac{1}{\alpha} \left(\mathbf{B}_i(\mathbf{x}_* + \alpha \mathbf{v}) - \mathbf{B}_i(\mathbf{x}_*) \right) . \quad (3.6)$$

Choosing $\alpha = \lambda \|\mathbf{z}\|$ with $\lambda > 0$ and for a sufficiently small λ, the above can be rewritten as:

$$\mathcal{D}_\mathbf{v} \mathbf{B}_i(\mathbf{x}_*) \simeq \frac{1}{\lambda \|\mathbf{z}\|} \left(\mathbf{B}_i(\mathbf{x}_* + \lambda \mathbf{z}) - \mathbf{B}_i(\mathbf{x}_*) \right) . \quad (3.7)$$

Plugging this and $\mathbf{v} = \mathbf{z}/\|\mathbf{z}\|$ into the expression $\nabla_{\mathbf{x}_*} \mathbf{B}_i(\mathbf{x}_*)^\top \mathbf{v} = \mathcal{D}_\mathbf{v} \mathbf{B}_i(\mathbf{x}_*)$ implies:

$$\nabla_{\mathbf{x}_*} \mathbf{B}_i(\mathbf{x}_*)^\top \mathbf{z} = \frac{1}{\lambda} \left(\mathbf{B}_i(\mathbf{x}_* + \lambda \mathbf{z}) - \mathbf{B}_i(\mathbf{x}_*) \right) . \quad (3.8)$$

As such, let $\ell_i(\mathbf{z}) \triangleq (\mathbf{z}/\lambda)(\mathbf{B}_i(\mathbf{x}_* + \lambda \mathbf{z}) - \mathbf{B}_i(\mathbf{x}_*))$,

$$\begin{aligned} \mathbb{E}\left[\ell_i(\mathbf{z})\right] &= \mathbb{E}\left[\mathbf{z} \nabla_{\mathbf{x}_*} \mathbf{B}_i(\mathbf{x}_*)^\top \mathbf{z}\right] \\ &= \mathbb{E}\left[\mathbf{z}\mathbf{z}^\top\right] \nabla_{\mathbf{x}_*} \mathbf{B}_i(\mathbf{x}_*) = \nabla_{\mathbf{x}_*} \mathbf{B}_i(\mathbf{x}_*) \end{aligned} \quad (3.9)$$

since $\mathbb{E}[\mathbf{z}\mathbf{z}^\top] = (\mathbb{V}[\mathbf{z}] + \mathbb{E}[\mathbf{z}]\mathbb{E}[\mathbf{z}]^\top) = \mathbf{I}$ due to $\mathbb{E}[\mathbf{z}] = 0$ and $\mathbb{V}[\mathbf{z}] = \mathbf{I}$ by definition. Thus, $\ell_i(\mathbf{z})$ is an unbiased stochastic gradient of $\nabla_{\mathbf{x}_*}\mathbf{B}_i(\mathbf{x}_*)$, which can then be estimated via averaging $\ell_i(\mathbf{z})$ over i.i.d. samples of $\mathbf{z} \sim \mathbb{N}(\mathbf{0}, \mathbf{I})$.

4 Experiments

To demonstrate the effectiveness of the proposed black-box model fusion via repurposing or reprogramming (**MFR**) approach, we set up experimental scenarios where multiple specialized pre-trained models need to be combined and reprogrammed to solve another related task with limited domain data. These are derived from real-world benchmark datasets in computer vision, which include MNIST (LeCun, Cortes, and Burges 2010), CIFAR-10 (Krizhevsky 2009), Mini-ImageNet (Vinyals et al. 2017) and the Large-Scale CelebFaces Attributes (CelebA) (Liu et al. 2015). We compare our approach against the following baselines:

Base Model: We train a new model from scratch using the few-shot dataset (few examples per class). Its parameterization (except for the final softmax prediction head) is identical to those of the other black boxes.

Black-Box Ensemble (BE): An ensemble of frozen black-box models. We concatenate their output and map the result to a softmax prediction on the target domain using a learnable feed-forward net with ReLU activation, appended with a final softmax transformation.

Fine-tuned Black-Box Ensemble (FBE): An ensemble of black-box models with tunable pre-softmax layers. We concatenate their outputs and pass the result through a learnable single-layer feed-forward net and a softmax transformation that maps it to the target prediction.

Black-Box Adversarial Re-programmer (BAR). We re-implemented the reprogramming framework from Tsai, Chen, and Ho (2020) to the best of our ability. This is because the released code for **BAR** is hard-coded for a binary classification task and a single black-box source, and cannot be used *as is* for our experiments. Their framework simultaneously learns to add adversarial noise to the input images and map the black-box outputs to new domains.

The numbers of learnable parameters in **BE** and **FBE** are comparable to that of our reprogramming model. The number of learnable parameters in Tsai, Chen, and Ho (2020) scales with the input size and cannot trivially be made comparable to our model. We used two invertible flow blocks, each composes of $\mathrm{PlanarFlow}$ and $\mathrm{RadialFlow}$ (Rezende and Mohamed 2016) in the implementation of our method. Last, we evaluate and compare all methods in the following settings: (1) in-domain repurposing; (2) cross-domain repurposing which are detailed below.

4.1 In-Domain Repurposing

For in-domain repurposing experiments, we pre-trained a number of black-box models that solve the classification task with different subsets of the classes of a single dataset. Then, given the corresponding black-box models, we use the above baselines (including our proposed algorithm) to repurpose them into a single model that can solve an unseen classification task with only a few examples per class. The above experiment is setup using the MNIST, CIFAR-10, Mini-ImageNet and CelebA datasets as detailed below.

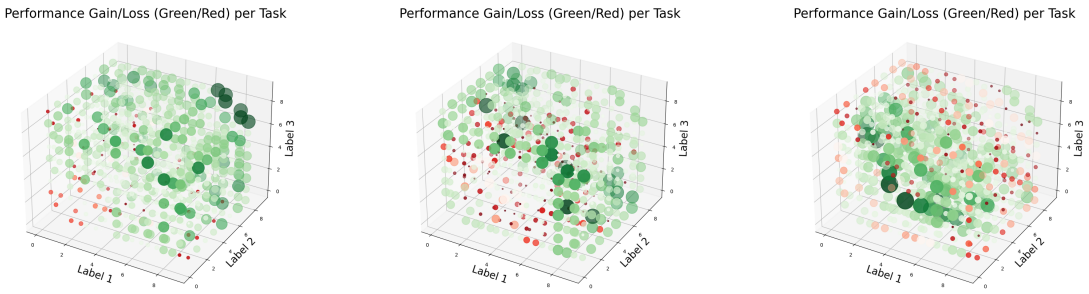
MNIST Experiment. The MNIST dataset comprises $60,000$ images of handwritten digits from 0 to 9. Among which $50,000$ images are used as training data and the remaining is used for testing (LeCun, Cortes, and Burges 2010). The training data is evenly distributed among the digits where each digit has 5000 training examples. To set up the experiment, we pre-trained 3 black-box models solving 3 separate MNIST sub-tasks which require them to distinguish between a set of 4 digits. These tasks are, respectively, $[1, 2, 3, 4]$, $[4, 5, 6, 7]$ and $[7, 8, 9, 0]$. Given the corresponding black-box models that solve these tasks, we use the above baselines to repurpose them into a single model that solves another MNIST 3-way classification task with 10 shots of data per class.

CIFAR-10 Experiment. The CIFAR-10 dataset consists of $60,000$ color images in 10 classes, with 6000 images per class (Krizhevsky 2009) . There are $50,000$ training images and $10,000$ test images. Similar to the MNIST experiments, we also built 3 black-box models solving the following classification tasks $(\mathrm{automobile}, \mathrm{bird}, \mathrm{cat}, \mathrm{deer})$, $(\mathrm{deer}, \mathrm{dog}, \mathrm{frog}, \mathrm{horse})$ and $(\mathrm{horse}, \mathrm{ship}, \mathrm{truck}, \mathrm{airplane})$, respectively, on the entire training dataset. Then, given these black-box models, we use the above baselines to repurpose them into a single model that solves another CIFAR-10 3-way classification task with 10 shots of data per class.

Mini-ImageNet Experiment. The Mini-ImageNet dataset comprises 100 classes with 600 samples of 84×84 color images per class (Vinyals et al. 2017). We sampled 10 classes and re-indexed them with labels in $[0, 9]$). Next, we built 3 black-box models solving 3 separate sub-tasks which require them to distinguish between a set of 4 classes. These are, respectively, $[1, 2, 3, 4]$, $[4, 5, 6, 7]$ and $[7, 8, 9, 0]$. Similar to the above, we repurpose these pre-trained models into a dedicated model to distinguish between image classes of another 3-way classification task. For each dataset, we report the repurposing results for all 3-way classification tasks. Given that there are 10 classes per dataset, there are a total number of 120 3-way classification tasks for each dataset. The results are reported in Fig. 2 above, which demonstrate the high success rate of repurposing:

1. The green bubbles correspond to target tasks for which our repurposed model outperforms the base model (success cases). Both the sizes and color gradients of these bubbles are set to be proportionate to the relative performance gain in the corresponding cases. Darker colors indicating more improvement over the base model.

2. Conversely, the red bubbles correspond to tasks where the repurposed models fail to improve over the base model (failure cases). Again, both sizes and color gradients of these bubbles are set to be proportionate to the relative performance drop in the corresponding cases. Observing Fig. 2, it can be seen that the size and number of green bubbles are significantly larger than those of red bubbles (across all datasets). This implies the no. of success cases is far larger than no. of failure cases, which speaks to the high success rate of repurposing. Although the reported few-shot performance of our model repurposing technique appears weaker



(a) MNIST (b) CIFAR-10 (c) Mini-ImageNet

Figure 2: Relative performance gain/loss (green/red) of our repurposed model over a base model on 120 3-way classification tasks in (a) MNIST, (b) CIFAR-10, and (c) Mini-ImageNet. Green and red bubbles denote target tasks where the repurposed model respectively outperforms and underperforms the base model. Darker color indicates a larger performance gap.

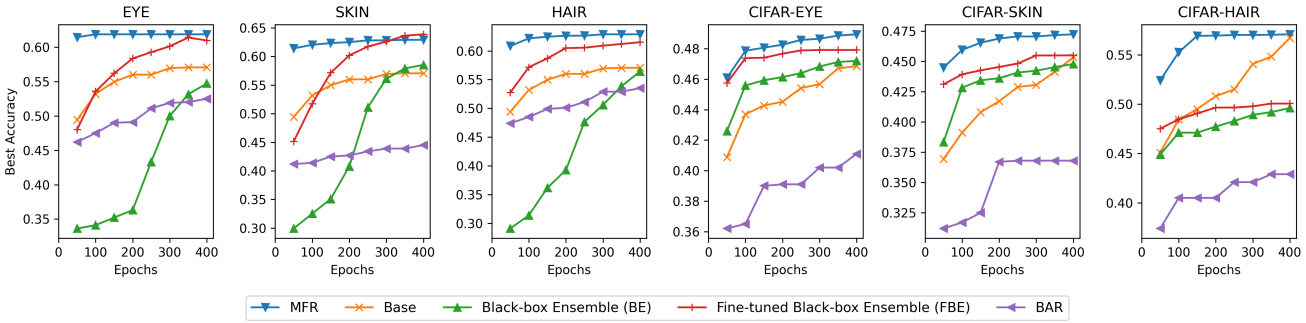


Figure 3: (left): Averaged repurposing performance from (a) EYE models; (b) SKIN models; and (c) HAIR models on a number of unseen target tasks. The average is over all target tasks generated under EYE, SKIN and HAIR categories; (right): Repurposing performance achieved by an ensemble of three CIFAR-10 black-box models on a randomly sampled target task under each CelebA sub-domain, including (d) EYE, (e) SKIN and (f) HAIR.

than state-of-the-art few-shot results for the above benchmark in standard meta learning setup, we emphasize that our setup is significantly more difficult since we have no access to labeled training data of the black-box models. Instead, standard meta learning approaches tend to be able to train their meta models on a large amount of labeled data from related tasks, which induces a stronger prior on data representation that leads to better few-shot performance.

4.2 Repurposing across Different Sub-Domains

Next, we further examine another form of in-domain repurposing in which the black-box models and the repurposing tasks were derived from different sub-domains of the same dataset. We demonstrate this on the CelebA dataset (Liu et al. 2015). We partition the CelebA dataset into 40 overlapping subsets of data. Each subset contains all training/test images that belong to one particular label (out of 40 labels in total). We consider data from subsets that belong to the following 3 categories: (1) **HAIR** features (9 labels); (2) **SKIN** features (5 labels); (3) **EYE** features (5 labels). From each category, we randomly sample four 3-way classification tasks. Three of which are source tasks and the remaining task is designated as the target task and given a 10-shot dataset. We repurpose the black-box models obtained from the training data of source tasks to solve target tasks from all categories. The averaged (in-domain) cross-category repurposing performance from each group (**HAIR**, **EYE** and **SKIN**) is reported in Figures 3a and 3. From all source categories, the (few-shot) repurposing performance of our proposed algorithm is significantly better than those of the other ensemble baselines as well as that of the base model. This is expected and consistent with our earlier observation in the MNIST, CIFAR-10 and Mini-ImageNet experiments. In addition, we also noted that the plotted results show that adversarial model repurposing performs the worst among all baselines. This is not surprising since Tsai, Chen, and Ho (2020) is grounded in the setting of a single pre-trained model setup which assumes that the black-box model has already internalized sufficient information to solve the target task. We however consider settings where each pre-trained model alone does not have enough information to solve the new task. Hence, combining them is necessary and can be

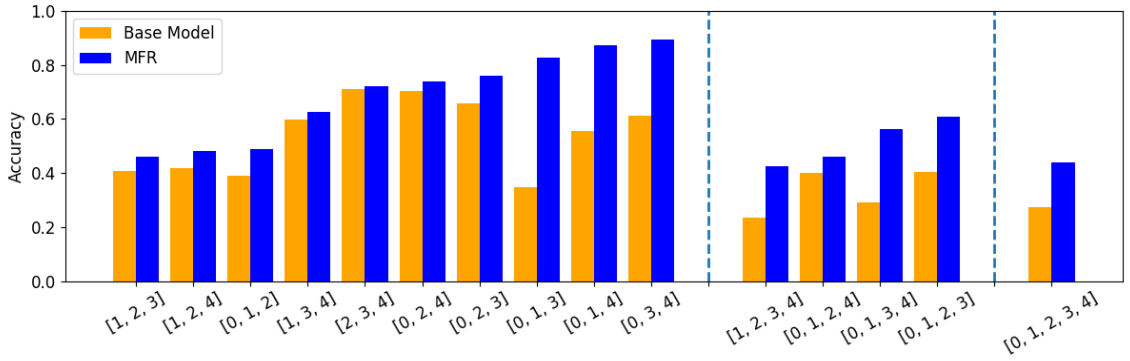


Figure 4: Plots of performance comparison between the base and repurposed models across all 3-way, 4-way and 5-way few-shot classification tasks. For each task, both the base and repurposed models are generated using the same few-shot dataset that contains 20 shots per class. Unlike the base models, the repurposed models also incorporated inferential insights from existing 2-way black-box models on the same set of classes which were pre-trained using the entire training dataset.

achieved via the following contributions. This necessitates more efficient exploitation of the black-box artifacts.

4.3 Cross-Domain Repurposing

The effectiveness of our repurposing algorithm can also be demonstrated across different data domains. In particular, we conduct the following experiments which repurpose (a) black-box classification models solving CIFAR-10 tasks to solve a target task under the categories of HAIR, EYE and SKIN of the CelebA dataset as described above; and (b) a ResNet-18 model trained on ImageNet data to solve a medical imaging classification problem derived from the Retinopathy Diabetic dataset, which is described below.

CIFAR-10 to CelebA Experiment. For this experiment, we use the same CIFAR-10 black-box models that were generated in the aforementioned CIFAR-10 experiment. These black-box models are then repurposed to solve the target tasks generated under the categories of **HAIR**, **EYE** and **SKIN** of the CelebA experiment above. The averaged repurposing performance over target tasks under each of these categories are plotted in Figures 3d and 3f above. All results consistently show improved repurposing performance of our proposed method over those of the other baselines. Additionally, we observe that adversarial model repurposing performs the worst among all baselines. This is consistent with our earlier observations for in-domain re-purposing.

ImageNet to Retinopathy-Diabetic Experiment. We down-sampled the Retinopathy Diabetic dataset to 10K images of retinas, labeled with the severity of the condition on a scale of 0 to 4 (larger is more severe). We reserved 50% of the data for testing and used the remaining to pre-train ResNet-18 black-box models for all 2-way tasks. We then show that selected ensembles of these black-box models can be repurposed into solutions to all 20-shot 3-way classification tasks, which are significantly more effective than their corresponding built-from-scratch solutions.

For each 3-way classification task (u, v, w) where $u < v < w \in \{0, 1, 2, 3, 4\}$, we repurpose an ensemble of 2-way black-box models, including those for classification tasks (u, v), (u, w) and (v, w), to solve it. For each 4-way task (u, v, p, q), which requires to distinguish between class labels u, v, p and q where $u < v < p < q \in \{0, 1, 2, 3, 4\}$, we repurpose a subset of 2-way black-box models $(a, b) \subset \{u, v, p, q\}$ to solve it. Finally, for the 5-way task $(0, 1, 2, 3, 4)$ which concerns all labels, we repurpose the following 3-way black-box models $\{(0, 1, 2), (0, 2, 3), (0, 3, 4)\}$ to solve it. We compare all repurposed models with their corresponding scratch models that were solely trained on the 20-shot datasets. All results reported in Figure 4 above consistently show that via repurposing, hidden knowledge within existing pre-trained black-box models can be unlocked and adapted effectively towards a wide range of few-shot classification tasks, showing significant improvement across all test cases. This reinforces and corroborates our earlier reported results on the sandbox datasets of MNIST, CIFAR-10 and Mini-ImageNet.

5 Conclusion

To complement previous black-box fine-tuning works that assume the existence of a large model pre-trained on generic data, our approach focuses on settings where multiple pre-trained models solving related tasks are available but none of which alone has sufficient knowledge to solve a target task. The core principle that underlies our approach is that the output of a related black-box models on an appropriately augmented (target) input can be leveraged into additional features that enhance the solution quality on few-shot learning (target) task. Our repurposing framework is developed to optimize such input augmentation functions so that their induced black-box output is most informative of the target label. Our empirical studies have demonstrated the feasibility and initial successes of the proposed approach in a variety of (cross-domain) few-shot learning tasks.

References

Baxter, J. 2000. A model of inductive bias learning. *Journal of artificial intelligence research*, 12: 149–198.

Ben-David, S.; and Schuller, R. 2003. Exploiting task relatedness for multiple task learning. In *Learning theory and kernel machines*, 567–580. Springer.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35: 1798–1828.

Bonilla, E. V.; Chai, K. M. A.; and Williams, C. K. I. 2008. Multi-task Gaussian Process Prediction. In *Proc. NIPS*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dinh, C. T.; Nguyen, T.; and Nguyen, T. D. 2020. Personalized Federated Learning with Moreau Envelopes. In *Proc. NeurIPS*.

Dinh, L.; Krueger, D.; and Bengio, Y. 2015. NICE: Non-linear Independent Components Estimation. arXiv:1410.8516.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2017. Density estimation using Real NVP. arXiv:1605.08803.

Du, S. S.; Hu, W.; Kakade, S. M.; Lee, J. D.; and Lei, Q. 2020. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized Federated Learning: Model-Agnostic Meta-Learning Approach. In *Proc. NeurIPS*.

Fifty, C.; Amid, E.; Zhao, Z.; Yu, T.; Anil, R.; and Finn, C. 2021. Efficiently Identifying Task Groupings for Multi-Task Learning. *arXiv preprint arXiv:2109.04617*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proc. ICML*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

Hoang, Q. M.; Hoang, T. N.; Low, K. H.; and Kingsford, C. 2019a. Collective Model Fusion for Multiple Black-Box Experts. In *Proc. ICML*.

Hoang, T. N.; Hoang, Q. M.; Low, K. H.; and How, J. P. 2019b. Collective Online Learning of Gaussian Processes in Massive Multi-Agent Systems. In *Proc. AAAI*.

Hoang, T. N.; Lam, C. T.; Low, K. H.; and Jaillet, P. 2020. Learning Task-Agnostic Embedding of Multiple Black-Box Experts for Multi-Task Model Fusion. In *Proc. ICML*.

Kestelman, H. 1961. Change of Variable in Riemann Integration. *The Mathematical Gazette*, 45(351): 17–23.

Kingma, D.; and Welling, M. 2013. Auto-Encoding Variational Bayes. In *Proc. ICLR*.

Kingma, D. P.; and Dhariwal, P. 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. arXiv:1807.03039.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, Department of Computer Science.

Lam, C. T.; Hoang, T. N.; Low, K. H.; and Jaillet, P. 2021. Model Fusion for Personalized Learning. In *Proc. ICML*.

LeCun, Y.; Cortes, C.; and Burges, C. 2010. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Pan, S. J.; and Yang, Q. 2009. A survey on transfer learning. *Transactions on Knowledge and Data Engineering*, 22(10): 1345–1359.

Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; and Huang, X. 2020. Pre-trained Models for Natural Language Processing: A Survey. *CoRR*, abs/2003.08271.

Rezende, D. J.; and Mohamed, S. 2016. Variational Inference with Normalizing Flows. arXiv:1505.05770.

Shao, J.-J.; Cheng, Z.; Li, Y.-F.; and Pu, S. 2021. Towards Robust Model Reuse in the Presence of Latent Domains. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, 2957–2963.

Tripuraneni, N.; Jordan, M. I.; and Jin, C. 2020. On the theory of transfer learning: The importance of task diversity. *arXiv preprint arXiv:2006.11650*.

Tsai, Y.-Y.; Chen, P.-Y.; and Ho, T.-Y. 2020. Transfer Learning without Knowing: Reprogramming Black-box Machine Learning Models with Scarce Data and Limited Resources. In *Proc. ICML*, 9614–9624.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2017. Matching Networks for One Shot Learning. arXiv:1606.04080.

Wang, Z.; Wang, X.; Shen, L.; Suo, Q.; Song, K.; Yu, D.; Shen, Y.; and Gao, M. 2022. Meta-learning without data via Wasserstein distributionally-robust model fusion. In Cussens, J.; and Zhang, K., eds., *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, 2045–2055. PMLR.

Wu, X.; Xu, W.; Liu, S.; and Zhou, Z. 2021. Model Reuse with Reduced Kernel Mean Embedding Specification. *TKDE*.

Wu, X.-Z.; Liu, S.; and Zhou, Z.-H. 2019. Heterogeneous Model Reuse via Optimizing Multiparty Multiclass Margin. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 6840–6849. PMLR.

Yoon, J.; Kim, T.; Dia, O.; Kim, S.; Bengio, Y.; and Ahn, S. 2018. Baysian Model-Agnostic Meta-Learning. In *Proc. NeurIPS*.