Learning Task-Agnostic Embedding of Multiple Black-Box Experts for Multi-Task Model Fusion

Trong Nghia Hoang [*1] **Chi Thanh Lam** [*2] **Bryan Kian Hsiang Low** [2] **Patrick Jaillet** [3]

Abstract

Model fusion is an emerging study in collective learning where heterogeneous experts with private data and learning architectures need to combine their black-box knowledge for better performance. Existing literature achieves this via a local knowledge distillation scheme that transfuses the predictive patterns of each pre-trained expert onto a white-box imitator model, which can be incorporated efficiently into a global model. This scheme however does not extend to multi-task scenarios where different experts were trained to solve different tasks and only part of their distilled knowledge is relevant to a new task. To address this multi-task challenge, we develop a new fusion paradigm that represents each expert as a distribution over a spectrum of predictive prototypes, which are isolated from task-specific information encoded within the prototype distribution. The task-agnostic prototypes can then be reintegrated to generate a new model that solves a new task encoded with a different prototype distribution. The fusion and adaptation performance of the proposed framework is demonstrated empirically on several real-world benchmark datasets.

1. Introduction

In various disciplines such as environmental sensing (Low et al., 2007; 2009; Podnar et al., 2010; Zhang et al., 2017), traffic monitoring (Hoang et al., 2014a;b; 2015; 2017; Yu et al., 2019) and healthcare analytics (Fu et al., 2019; Hong et al., 2019; Xiao et al., 2019; Huang et al., 2020), observational data that describe the same phenomenon or concept are often acquired from multiple experiments, which are conducted on different subjects. The data that they generated would therefore have different distributions or statistical properties. In practice, due to privacy concerns, data collected from different acquisition frameworks (e.g., sensors and/or experiments) are also private and cannot be shared among themselves (McMahan et al., 2017; Yoon et al., 2018; Hoang et al., 2019b). This creates private datasets of the same phenomenon, where each is used to train a separate model from a local perspective. For example, in clinical research, patient information is often recorded across different institutions (Xu & Wang, 2019), which do not share data with each other due to protect the patient's sensitive information. As such, each institution can only model the patient population using data collected from a single demographic region, which might not generalize well to others.

Furthermore, in settings with strict security requirements, parameters of a local model also need to be kept private due to a recently discovered threat of adversarial ML attack (Finlayson et al., 2019; Zhao et al., 2019), which essentially makes it a black box to others. This violates the model transparency requirement of existing distributed modeling works addressing this data federation issue (McMahan et al., 2017; Hoang et al., 2019b; Yurochkin et al., 2019a;b; Singh & Jaggi, 2019), which results in the black-box challenge.

This led us to the recent development of a black-box fusion framework (Hoang et al., 2019a) which allows multiple agents (e.g., medical institutions) to transfuse their black-box knowledge onto the corresponding white-box surrogates that share the same model architecture. Each surrogate can then exchange and aggregate the parameter gradients of their predictions to generate a parameter-correcting gradient that combines their learning priors.

However, similar to the existing literature in federated learning (McMahan et al., 2017; Hoang et al., 2019b; Yurochkin et al., 2019a;b), this work assumes local models were trained to solve the same task (albeit with different data). Their proposed methods in fact do not isolate task-specific (irrelevant knowledge) from task-agnostic information (relevant knowledge) which are implicitly entangled in the parameter representation of each model. This entanglement presents a problem in multi-task setting since it does not tell us which part of an existing model is relevant to a new task and which part is not. This is also an issue in a remotely relevant litera-

[*]Equal contribution [1]MIT-IBM Watson AI Lab [2]National University of Singapore [3]Massachusetts Institute of Technology. Correspondence to: Trong Nghia Hoang <nghiaht@ibm.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

ture of meta learning (Finn et al., 2017; Yoon et al., 2018) which tackles the model adaptation problem in multi-task scenarios from a different technical setting that does not accommodate black-box and pre-trained models with private training data. For interested readers, a succinct review of meta learning is provided in Section 2 below.

Motivated by the above intuition, this paper investigates a different fusion paradigm that represents each black-box expert as a task-dependent distribution over an infinite spectrum of task-agnostic predictive prototypes. The prototypes intuitively represent the task-agnostic information, which can be transferred among tasks to corroborate their statistical strength, whereas their distribution encodes domain-specific information that needs to be adapted to suit a new task. This is substantiated by the following technical contributions:

Disentangled Black-Box Embedding. We leverage both the meta information of each task and observations of its model's prediction outcome at a subset of unlabeled training data to compute an embedded representation for these black boxes. The embedding factorizes across two separate latent sub-spaces wherein one encodes generic input patterns (e.g., basic features of a face) while the other encodes conceptual patterns (e.g., specific facial expressions) that are correlated with the black-box's prediction. A task, such as emotion prediction for a certain group of people, can then be represented as a distribution over the generic patterns (Section 3.1).

Task-Agnostic Decomposition. We exploit the developed disentangled representation to induce a natural decomposition for an arbitrary model, which is expressed as an integration over a spectrum of task-agnostic prototypes. This allows the task-specific information to be succinctly isolated within a low-complexity parameterization of the integration distribution, which can be easily adapted into a new task domain even with limited training examples (Section 3.2).

Model Fusion and Adaptation. We develop a new fusion and adaptation algorithm that learns to generate the above model embedding and adaptation of prototype distribution in an end-to-end fashion. This enables communication between the embedding and adaptation components, which allows them to converge on a latent representation that is optimized for both model reconstruction – how to recompose the decomposed prototypes to best reproduce each black box – and adaptation – how to adapt the prototype distribution to best suit a new task domain characterized by a few shots of training examples (Section 3.4).

To demonstrate the efficacy of our framework, we evaluate its performance in a wide range of settings and on multiple benchmarks, which include the MNIST (LeCun et al., 2010), n-digit MNIST (Oh et al., 2018) and Mini-ImageNet (Ravi & Larochelle, 2017) datasets. Our empirical studies essentially show that by using the proposed task-agnostic representation, the fused model is able to localize its adaptation to the most relevant parts of its prototypical representation, which can be adapted well to any new task with limited training data (Section 4).

2. Related Works

2.1. Model Fusion

Model fusion (Hoang et al., 2019a;b) is an emerging study that arises recently from the traditional context of distributed machine learning (ML) where a single analytic model is engineered in the cloud (Chen et al., 2013a; Low et al., 2015; Deisenroth & Ng, 2015; Hoang et al., 2016) as a service to be used by local machines. Distributed ML thus requires broadcasting data statistics from local experts to a central server for processing. These works, however, did not account for the privacy of data where local models cannot access each others' private data.

To accommodate this privacy constraint, existing works in distributed (Allamraju & Chowdhary, 2017; McMahan et al., 2017; Hoang et al., 2018a; 2019b; Yurochkin et al., 2019a;b) and/or multi-agent learning (Chen et al., 2012; 2013b; Hoang & Low, 2013; Ruofei & Low, 2018; Hoang et al., 2018b) enforce an identical knowledge representation across all experts to impose a (hierarchical) statistical relationship between their parameters. This enables efficient communication and aggregation of predictive knowledge among themselves via inferential computation. This is, however, not desirable in practical domains with extra proprietary constraints imposed on the models themselves where local experts cannot communicate in advance to agree on the same model architecture, and also do not want to communicate their model parameters due to a recently discovered threat of adversarial attack (Finlayson et al., 2019; Zhao et al., 2019). In contrast, allowing their architecture to be heterogeneous and to remain as black-box interface avoids these problems, but causes difficulty in communication.

To resolve this dilemma, the most recent work of (Hoang et al., 2019a) proposed a collective black-box fusion mechanism that allows black-box experts to interact, learn and distill the predictive behaviors of one another into white-box surrogates encoded with homogeneous information summaries. This allows the black boxes to represent, communicate and combine their expertise efficiently to harness the full potential of their collective intelligence without having to publicize their private data and/or model architecture. It does not however accommodate for black-box models that were trained to solve different (but related) tasks.

This necessitates the development of new computation capabilities to disentangle relevant knowledge from irrelevant knowledge that might carry misleading inductive bias, which is the focal theme of this paper (Section 3).

2.2. Meta Learning

Different from model fusion which combines pre-trained models solving the same task into a new model with improved performance, meta learning aims to compute a universal model initializer for a given distribution of task (or meta description of task) which can be adapted to any task characterized by its few shots of training examples.

More concretely, in meta-learning, each task is indexed with a task descriptor τ (e.g., the vector of coefficients that define it) and all tasks accept the same input and output spaces. The task descriptor is further assumed to be distributed by a given distribution $p(\tau)$. A model $Q(\mathbf{x};\gamma)$ is defined to be a mapping from an input $\mathbf{x}$ to an outcome in the output space, which is parameterized by γ. The performance of Q in task τ is then assessed by a differentiable loss function $\mathbf{L}_\tau(\gamma)$.

During the meta-learning phase, we want to learn γ_* for $Q(\mathbf{x};\gamma_*)$ such that it can be quickly updated to perform well in an arbitrary new task $\tau' \sim p(\tau)$ using only its K shots of training examples $\mathbf{D}_{\tau'} = \{\mathbf{x}^{(i)}_{\tau'}, y^{(i)}_{\tau'}\}_{i=1}^K$. Assuming that $Q(\mathbf{x};\gamma_*)$ is updated via gradient descent, meta learning aims to compute γ_* such that starting from γ_*, this rule can make rapid progress on any new task $\tau' \sim p(\tau)$. This is achieved via minimizing (Finn et al., 2017),

$$\gamma_* \quad = \quad \min_{\gamma} \mathbb{E}_{\tau\sim p(\tau)} \Big[\mathbf{L}_\tau\Big(\gamma - \alpha\nabla_\gamma \mathbf{L}_\tau(\gamma)\Big)\Big] \ , \qquad (1)$$

where α is a tunable step-size parameter. This in turn can be achieved via stochastic gradient descent with unbiased samples $\tau_1, \tau_2, \ldots, \tau_n \sim p(\tau)$:

$$\begin{aligned}\gamma_* \quad &= \quad \gamma_* - \beta \sum_{i=1}^n \nabla_\gamma \Big[\mathbf{L}_{\tau_i}\Big(\gamma_* - \alpha\nabla_\gamma \mathbf{L}_{\tau_i}(\gamma_*)\Big)\Big] \\ &= \quad \gamma_* - \beta \sum_{i=1}^n \Big[\Big(\mathbf{I} - \alpha\nabla^2_\gamma \mathbf{L}_{\tau_i}(\gamma_*)\Big)\nabla_\gamma \mathbf{L}_{\tau_i}(\gamma_i)\Big] \quad (2)\end{aligned}$$

where $\gamma_i = \gamma_* - \alpha\nabla_\gamma \mathbf{L}_{\tau_i}(\gamma_*)$, $\nabla^2_\gamma \mathbf{L}_{\tau_i}(\gamma_*)$ is the Hessian of $\mathbf{L}_{\tau_i}(\gamma)$ evaluated at γ_*, $\mathbf{I}$ is the identity matrix and β is a tunable meta-learning step size. For computational efficiency, first-order meta learning ignores the Hessian, which reduces the above update rule to the following alternating gradient adaptation and aggregation, $\gamma_i = \gamma_i - \alpha\nabla_\gamma \mathbf{L}_{\tau_i}(\gamma_*)$ and $\gamma_* = \gamma_* - \beta\sum_{i=1}^n \nabla_\gamma \mathbf{L}_{\tau_i}(\gamma_i)$.

2.3. Variational Auto-Encoder (VAE)

Our embedding component is built on the seminal variational auto-encoder (VAE) framework for **data embedding** of (Kingma & Welling, 2013). We review the basic of VAE below and discuss how to extend it towards a factorized **model embedding** later in Section 3.1.

Let $\mathbf{x}$ denote a random variable that is distributed according to an unknown density $p_D(\mathbf{x})$. We want to learn a latent variable model $p_\theta(\mathbf{x},\mathbf{z}) \triangleq p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$ that captures this generative process. The latent variable model comprises a fixed latent prior $p(\mathbf{z})$ and a parametric likelihood $p_\theta(\mathbf{x}|\mathbf{z})$. To learn θ, we maximize the variational evidence lower-bound (ELBO) $\mathbf{L}(\mathbf{x};\theta,\phi)$ of $\log p_\theta(\mathbf{x})$:

$$\mathbf{L}(\mathbf{x};\theta,\phi) \triangleq \mathbb{E}_{\mathbf{z}\sim q_\phi}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big] - \mathbb{KL}\Big(q_\phi(\mathbf{z}||\mathbf{x})||p(\mathbf{z})\Big)$$

with respect to an arbitrary posterior surrogate $q_\phi(\mathbf{z}|\mathbf{x}) \simeq p_\theta(\mathbf{z}|\mathbf{x})$ over the latent variable $\mathbf{z}$.

This can be viewed as a stochastic auto-encoder with $p_\theta(\mathbf{x}|\mathbf{z})$ and $q_\phi(\mathbf{z}|\mathbf{x})$ acting as the encoder and decoder, respectively. Here, θ and ϕ characterize the neural network parameterization of these models. Their learning is enabled via a re-parameterization of $q_\phi(\mathbf{z}|\mathbf{x})$ that enables stochastic gradient ascent (SGA) (Kingma & Welling, 2013).

3. Multi-Task Black-Box Model Fusion

Let $\mathbf{B}_{\tau_1}, \mathbf{B}_{\tau_2}, \ldots, \mathbf{B}_{\tau_p}$ denote the p expert models which were pre-trained to solve p related tasks $\tau_1, \tau_2, \ldots, \tau_p$ where each τ_i describes the corresponding task's meta information (e.g., attributes describing the data population and prediction target). Each model $\mathbf{B}_{\tau_i}$ acts as a black-box interface which returns a predictive distribution η over a label space for each input $\mathbf{x}$ (e.g., images) from an unlabeled set of data $\mathbf{U}$.

Given an unseen task with meta information τ_*, we are interested in learning a new model $\mathbf{B}_{\tau_*}$ that performs well on τ_* even if $\mathbf{D}_{\tau_*}$ only provides a few shots of examples. Assuming τ_* is drawn from the same distribution $p(\tau)$, this can be achieved by distilling task-agnostic knowledge from $\mathbf{B}_{\tau_1}, \mathbf{B}_{\tau_2}, \ldots, \mathbf{B}_{\tau_p}$, and recomposing them to best fit the provided few shots of training data. This defines the multi-task model fusion task that this paper aims to address.

3.1. Disentangled Black-Box Embedding

Let $\mathbf{H} = \mathbf{W} \times \mathbf{Z}$ denote a factored space that embeds the latent features describing both the task and its solution model. Our goal is to learn a latent representation that distills task-dependent information in $\mathbf{W}$, while encoding the rest within $\mathbf{Z}$. That is, $\mathbf{W}$ isolates generic input patterns from $\mathbf{Z}$ which encodes task-agnostic concepts that underline the black-box's inferential mechanism. Both $\mathbf{W}$ and $\mathbf{Z}$ contain encoding generative information of the input. As a concrete example, consider the problem of handwritten digits classification, we want to encourage the following behavior in our model: $\mathbf{z}$ encodes the information central to making predictions (i.e. the numerical value of the digit) whereas $\mathbf{w}$ encodes information that does not influence the prediction, such as the width of the stroke, the angle in which the digit is written, the light intensity of the images, and other abstract stylistic properties.

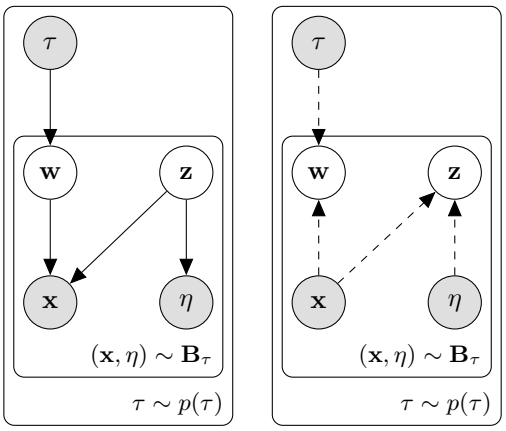


Figure 1. Graphical models of (a) the generative and (b) inference networks – $p(\mathbf{w}, \mathbf{z}, \mathbf{x}, \eta, \tau; \theta, \gamma, \alpha)$ and $q(\mathbf{w}, \mathbf{z}|\mathbf{x}, \eta, \tau; \phi)$, respectively – in our auto-encoding module. The dashed arrows in (b) indicate the posterior surrogates that form the inference network.

Under this modeling paradigm (see Fig. 1), we adopt the following parameterization for $p(\mathbf{w}, \mathbf{z}, \mathbf{x}, \eta, \tau; \theta, \gamma, \alpha)$,

$$p(\mathbf{w}, \mathbf{z}, \mathbf{x}, \eta, \tau; \theta, \gamma, \alpha) \triangleq p_\theta(\mathbf{x}|\mathbf{w}, \mathbf{z}) p_\gamma(\mathbf{w}|\tau) \; p_\alpha(\eta|\mathbf{z}) p(\tau) p(\mathbf{z}) \; , \tag{3}$$

where θ, γ, α denote an abstract parameterization often implemented in form of a (deep) neural network in our experiment. We assume the latent priors $p(\mathbf{z})$ and $p(\tau)$ are fixed and can be sampled from. Thus, learning this representation means learning (θ, γ, α) that best explains the observations $(\mathbf{x}, \eta, \tau)$ which were collected by observing the prediction of $\mathbf{B}_{\tau_1}, \mathbf{B}_{\tau_2}, \ldots, \mathbf{B}_{\tau_p}$ at the unlabeled data $\mathbf{x} \in \mathbf{D}$.

Concretely, this means optimizing for (θ, γ, α) that maximizes the following expected model evidence,

$$\underset{(\theta,\gamma,\alpha)}{\text{maximize}} \quad \mathbb{E}_{(\mathbf{x},\tau,\eta)\sim\mathbf{R}} \Big[\log p(\mathbf{x}, \eta, \tau; \theta, \gamma, \alpha) \Big] \; , \tag{4}$$

where $\mathbf{R}$ denotes the set of observations $(\mathbf{x}, \eta, \tau)$ and $(\mathbf{x}, \eta, \tau) \sim \mathbf{R}$ denotes the above expectation as an empirical average over $\mathbf{R}$. Optimizing Eq. (4) however is difficult since $\log p(\mathbf{x}, \eta, \tau; \theta, \gamma, \alpha)$ is intractable. To sidestep this issue, we instead exploit a surrogate distribution,

$$q_\phi(\mathbf{w}, \mathbf{z}|\mathbf{x}, \tau, \eta) \triangleq q_\phi(\mathbf{z}|\mathbf{x}, \eta) \; q_\phi(\mathbf{w}|\mathbf{x}, \tau) \tag{5}$$

to approximate the true posterior over the latent variable $p(\mathbf{w}, \mathbf{z}|\mathbf{x}, \tau, \eta; \theta, \gamma, \alpha)$. This is otherwise known (in the context of variational auto-encoder) as the inference network, which is parameterized by ϕ. Using $q_\phi(\mathbf{w}, \mathbf{z}|\mathbf{x}, \tau, \eta)$, we can re-express $\log p(\mathbf{x}, \eta, \tau; \theta, \gamma, \alpha)$ as

$$\begin{aligned} \log p(\mathbf{x}, \eta, \tau; \theta, \gamma, \alpha) &= \mathbf{L}(\mathbf{x}, \eta, \tau; \theta, \gamma, \alpha, \phi) + \mathbb{KL}(q_\phi||p) \\ &\geq \mathbf{L}(\mathbf{x}, \eta, \tau; \theta, \gamma, \alpha, \phi) \end{aligned} \tag{6}$$

where $\mathbb{KL}(q_\phi||p)$ is the short-hand notation for the Kullback-Leibler divergence $\mathbb{KL}(q_\phi(\mathbf{w}, \mathbf{z}|\mathbf{x}, \tau, \eta)||p(\mathbf{w}, \mathbf{z}|\mathbf{x}, \tau, \eta))$, and $\mathbf{L}(\mathbf{x}, \eta, \tau; \theta, \gamma, \alpha, \phi)$ denotes the individual variational lower-bound of $\log p(\mathbf{x}, \eta, \tau; \theta, \gamma, \alpha)$,

$$\mathbf{L}(\mathbf{x}, \eta, \tau; \theta, \gamma, \alpha, \phi) = \mathbb{E}_{q_\phi} \left[\log \frac{p(\mathbf{w}, \mathbf{z}, \mathbf{x}, \eta, \tau; \theta, \gamma, \alpha)}{q_\phi(\mathbf{w}, \mathbf{z}|\mathbf{x}, \tau, \eta)} \right] \tag{7}$$

which can be further decomposed as

$$\begin{aligned} \mathbf{L}(\mathbf{x}, \eta, \tau; \theta, \gamma, \alpha, \phi) = \mathbb{E}_{q_\phi} \Big[&\log p_\theta(\mathbf{x}|\mathbf{w}, \mathbf{z}) + \log p_\alpha(\eta|\mathbf{z}) \Big] \\ &- \mathbb{KL}\Big(q_\phi(\mathbf{z}|\mathbf{x}, \eta)||p(\mathbf{z}) \Big) + \log p(\tau) \\ &- \mathbb{KL}\Big(q_\phi(\mathbf{w}|\mathbf{x}, \tau)||p_\gamma(\mathbf{w}|\tau) \Big) \end{aligned} \tag{8}$$

Similar to the variational lower-bound of VAE (Section 2.3), Eq. (8) is expressed as an expectation over the surrogate instead of the true posterior over $(\mathbf{w}, \mathbf{z})$, which allows us to compute its unbiased stochastic gradient via re-parameterization (Kingma & Welling, 2013). Thus, we can learn (θ, γ, α) via optimizing $\mathbb{E}[\mathbf{L}(\mathbf{x}, \eta, \tau; \theta, \gamma, \alpha, \phi)]$ instead of the expected model evidence in Eq. (4).

3.2. Task-Agnostic Prototype Decomposition

Once (θ, γ, α) is learned, we can express the distribution over η conditioned on $\mathbf{x}$ and τ as follows,

$$\begin{aligned} p(\eta|\mathbf{x}, \tau) &\propto \int_{\mathbf{w}} \int_{\mathbf{z}} p(\mathbf{w}, \mathbf{z}, \mathbf{x}, \eta|\tau; \; \theta, \gamma, \alpha) \mathrm{d}\mathbf{z} \mathrm{d}\mathbf{w} \\ &= \int_{\mathbf{w}} G_{\mathbf{w}} \left(\eta|\mathbf{x}; \theta, \alpha \right) p_\gamma(\mathbf{w}|\tau) \mathrm{d}\mathbf{w} \; , \end{aligned} \tag{9}$$

where $G_{\mathbf{w}}(\eta|\mathbf{x}; \theta, \alpha)$ is expressed as an integration over $\mathbf{z}$,

$$\begin{aligned} G_{\mathbf{w}}(\eta|\mathbf{x}; \theta, \alpha) &= \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{w}, \mathbf{z}) p_\alpha(\eta|\mathbf{z}) p(\mathbf{z}) \mathrm{d}\mathbf{z} \\ &\simeq \frac{1}{m} \sum_{i=1}^{m} p_\theta(\mathbf{x}|\mathbf{z}_i, \mathbf{w}) p_\alpha(\eta|\mathbf{z}_i) \; , \end{aligned} \tag{10}$$

where the samples $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m$ are drawn independently and identically from the latent prior $p(\mathbf{z})$. This allows us to efficiently store the entire spectrum of $G_{\mathbf{w}}$ in memory a priori via sampling from $p(\mathbf{z})$ and caching the previously learned embedding parameterization (θ, α). The derivation of Eq. (9) and Eq. (10) above follows immediately from the marginalization and factorization in Eq. (3).

Then, by associating an arbitrary black-box model $\mathbf{B}_\tau(\mathbf{x}) \equiv \arg\max_\eta p(\eta|\mathbf{x}, \tau)$ with the above distribution over prediction outcome, Eq. (9) reveals a decomposition of $\mathbf{B}_{\tau_1}, \mathbf{B}_{\tau_2}, \ldots, \mathbf{B}_{\tau_p}$ into a spectrum of task-agnostic inferential prototypes $G_{\mathbf{w}}(\eta|\mathbf{x}; \theta, \alpha)$ succinctly characterized by (θ, α). Note that while $G_{\mathbf{w}}(\eta|\mathbf{x}; \theta, \alpha)$ is structured around a generic pattern $\mathbf{w}$ – see Eq. (10), it is not specific to any task

τ which is characterized by a different prototype distribution $p_\gamma(\mathbf{w}|\tau)$ parameterized by γ.

Intuitively, $G_{\mathbf{w}}(\eta|\mathbf{x};\theta,\alpha)$ encodes inferential knowledge that is commonly shared across tasks while $p_\gamma(\mathbf{w}|\tau)$ captures information specific to τ which measures the relevance of each inferential prototype $G_{\mathbf{w}}(\eta|\mathbf{x};\theta,\alpha)$ to τ. This presents an explicit construction of $\mathbf{B}_{\tau_*}$ which maps from input $\mathbf{x}$ to the most likely outcome η,

$$\begin{aligned}\mathbf{B}_{\tau_*}(\mathbf{x}) &= \int_{\mathbf{w}} \Big[\arg\max_\eta G_{\mathbf{w}}(\eta|\mathbf{x};\theta,\alpha)\Big] p_\gamma(\mathbf{w}|\tau_*)\mathrm{d}\mathbf{w}\,, \\ &= \mathbb{E}_{\mathbf{w}}\Big[\arg\max_\eta G_{\mathbf{w}}(\eta|\mathbf{x};\theta,\alpha)\Big]\,, \end{aligned} \tag{11}$$

where the expectation is over $\mathbf{w} \sim p_\gamma(\mathbf{w}|\tau_*)$ and as such, the agnostic prototypes $G_{\mathbf{w}}(\eta|\mathbf{x};\theta,\alpha)$ are recomposed via task τ_*'s prototype preference $p_\gamma(\mathbf{w}|\tau_*)$. Since the integration in Eq. (11) is generally intractable, we instead replace it by the following unbiased estimate:

$$\mathbf{B}_{\tau_*}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\Big[\arg\max_\eta G_{\mathbf{w}_i}(\eta|\mathbf{x};\theta,\alpha)\Big]\,, \tag{12}$$

where the samples $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n$ are drawn identically and independently from $p_\gamma(\mathbf{w}|\tau_*)$.

Eq. (10) and Eq. (12) thus characterize two important steps in our new fusion paradigm: (1) **decomposition** to distill task-agnostic inferential prototypes from existing models solving related tasks – see Eq. (10); and (2) **recomposition** to re-integrate them to solve a new task drawn from the same task distribution – see Eq. (12).

Issues. There are however two remaining issues that need to be addressed (see Sections 3.3 and 3.4):

1. Though our modeling in Section 3.1 stipulates that the latent $\mathbf{z}$ that captures only generic input patterns would have no influence on the prediction η of the black-box, there is no explicit mechanism to enforce it. Intuitively, we need to make sure changing the distribution of $\mathbf{z}$ would not change the distribution of η if $\mathbf{x}$ is observed;

2. The recomposed model $\mathbf{B}_{\tau_*}$ only fits what the generative scheme in Section 3.1 believes to be data that were plausibly generated from task τ_* via observing $\mathbf{B}_{\tau_1}, \mathbf{B}_{\tau_2}, \ldots, \mathbf{B}_{\tau_p}$. Intuitively, this is a fit *on average* over an entire data distribution induced by (θ,α,γ), which might not fit well at a particular set $\mathbf{D}_{\tau_*}$ that contains a few shots of training examples for an unseen task τ_*.

Addressing these issues, as explained in Sections 3.3 and 3.4 below, will show how the above algorithmic components in Sections 3.1 and 3.2 can be put together in an end-to-end training pipeline where (intuitively) the need to suit a certain task is back-propagated to guide the embedding process towards a particular decomposition that can be best recomposed to solve the task.

3.3. Decoupling Task-Agnostic and Task-Dependent Variables via Minimizing Mutual Information

To address the first issue, we need to regularize our embedding loss in Eq. (4) such that it prioritizes representation that induces minimal mutual information between $\mathbf{w}$ and η. Since (θ,γ,α) does not have an explicit influence on the mutual information between $\mathbf{w}$ and η, we need to introduce an extra surrogate component that captures this and interact with (θ,γ,α) at the same time.

Concretely, this is achieved by expressing the mutual information as $\mathcal{I}(\eta;\mathbf{w}) = \mathbb{H}(\eta) - \mathbb{H}(\eta|\mathbf{w})$ in which the first term is a constant and can be ignored. Minimizing $\mathcal{I}(\eta;\mathbf{w})$ is therefore equivalent to minimizing $-\mathbb{H}(\eta|\mathbf{w})$

$$\begin{aligned}&= \int_{\eta,\mathbf{w}} p(\eta,\mathbf{w})\log p(\eta|\mathbf{w})\mathrm{d}\eta\mathrm{d}\mathbf{w} \\ &= \mathbb{E}_{p(\mathbf{x},\tau,\eta)}\left[\int_{\mathbf{w},\mathbf{z}} p(\mathbf{w},\mathbf{z}|\mathbf{x},\tau,\eta)\log p(\eta|\mathbf{w})\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{z}\right] \\ &\simeq \mathbb{E}_{(\mathbf{x},\tau,\eta)\sim\mathbf{R}}\left[\int_{\mathbf{w},\mathbf{z}} q_\phi(\mathbf{w},\mathbf{z}|\mathbf{x},\tau,\eta)\log q_\psi(\eta|\mathbf{w})\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{z}\right] \end{aligned} \tag{13}$$

where the first equality follows from the marginalization identity of $p(\eta,\mathbf{w})$ over $(\mathbf{x},\tau,\eta)$. It is then followed by three approximations in the next step: (a) approximate the true marginal $p(\mathbf{x},\tau,\eta)$ by its empirical distribution via $\mathbf{R}$ as defined after Eq. (4); (b) re-use $q_\phi(\mathbf{w},\mathbf{z}|\mathbf{x},\tau,\eta)$ in Eq. (6) to approximate the true posterior; and (c) approximate $p(\eta|\mathbf{w})$ by introducing a new surrogate $q_\psi(\eta|\mathbf{w})$.

This in turn enables minimizing $\mathcal{I}(\eta;\mathbf{w})$ via maximizing $\mathbb{E}[\mathbf{F}_{\mathcal{I}}(\mathbf{x},\tau,\eta;\phi,\psi)]$ where the expectation is over $(\mathbf{x},\tau,\eta) \sim \mathbf{R}$ and the **information regularizer** $\mathbf{F}_{\mathcal{I}}(\mathbf{x},\tau,\eta;\phi,\psi)$ is given as

$$\begin{aligned}\mathbf{F}_{\mathcal{I}}(\mathbf{x};\tau,\eta;\phi,\psi) &= \int_{\mathbf{w},\mathbf{z}} q_\phi(\mathbf{w},\mathbf{z}|\mathbf{x},\tau,\eta)\log q_\psi(\eta|\mathbf{w})\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{z} \\ &= \int_{\mathbf{w}} q_\phi(\mathbf{w}|\mathbf{x},\tau)\log q_\psi(\eta|\mathbf{w})\mathrm{d}\mathbf{w} \end{aligned} \tag{14}$$

Intuitively, $q_\psi(\eta|\mathbf{w})$ is the new surrogate component that captures the information relationship between η and $\mathbf{w}$, which interestingly interacts with ϕ via $\mathbf{F}_{\mathcal{I}}(\phi,\psi)$ and with (θ,γ,α) by extension if we optimize $\mathbf{L}(\mathbf{x},\tau,\eta;\theta,\gamma,\alpha,\phi)$ and $\mathbf{F}_{\mathcal{I}}(\mathbf{x},\tau,\eta;\phi,\psi)$ together.

3.4. Model Fusion and Adaptation via Minimizing PAC-Bayesian Generalization Bound

To address and further elaborate on the second issue, recall that the decomposed prototypes $G_{\mathbf{w}}(\eta|\mathbf{x};\theta;\alpha)$ in Section 3.2 are task-independent. Thus, to incorporate them into a new model $\mathbf{B}_{\tau_*}$ that solves an unseen task τ_*, one approach is to re-integrate them via Eq. (11) with respect to τ_*'s prototype distribution $p_\gamma(\mathbf{w}|\tau_*)$ that (intuitively) expresses τ_*'s *prototype preference*. However, while such

distribution can be immediately induced from the learned generative model in Section 3.1, its parameterization γ, which was learned as part of the aforementioned generative model, does not take into account the few shots of training examples $\mathbf{D}_{\tau_*}$ of the new task τ_*. As such, $p_\gamma(\mathbf{w}|\tau_*)$ needs to be further adapted to fit $\mathbf{D}_{\tau_*}$ and one principled method to achieve this is to view $p_\gamma(\mathbf{w}|\tau_*)$ as a reference *prior* over a space of hypotheses $G_{\mathbf{w}}$ which were established without observing $\mathbf{D}_{\tau_*}$. One might then attempt to use Bayes rule to compute a posterior over these hypotheses given the prior $p_\gamma(\mathbf{w}|\tau_*)$ and the observations $\mathbf{D}_{\tau_*}$. However, this approach does not work unless we have already had access to the likelihood of observing $\mathbf{D}_{\tau_*}$ from $\mathbf{B}_{\tau_*}$ which is what needs to be estimated in the first place.

To resolve this, we instead adopt the PAC-Bayes method (McAllester, 1999; Shalev-Shwartz & Ben-David, 2014), which allows one to provably learn a posterior that best fits the observations without knowing their likelihood. Instead, this is achieved by re-characterizing the observation fitness via a parameterized Gibbs loss,

$$\mathbf{G}\left(q_\lambda\right) \;=\; \mathbb{E}_{(\mathbf{x},y)\sim\pi}\left[\mathbb{E}_{\mathbf{w}\sim q_\lambda}\left[\ell\Big(\eta_{\mathbf{w}}(\mathbf{x}), y\Big)\right]\right], \qquad (15)$$

where π is the latent data distribution of τ_* unbeknownst to us, $\eta_{\mathbf{w}}(\mathbf{x}) = \arg\max_\eta G_{\mathbf{w}}(\eta|\mathbf{x};\theta,\alpha)$ and $\ell(\eta, y) = 1 - (\eta)_y$ where $(\eta)_y$ is the y-th component of η. In addition, the above loss measure in (15) is also parameterized by λ which defines a surrogate posterior q_λ over the space of prototypes $G_{\mathbf{w}}$. Learning λ that minimizes $\mathbf{G}(q_\lambda)$ then produces the desired posterior $q_\lambda(\mathbf{w})$ that fit the observations. That said, a direct minimization of $\mathbf{G}(q_\lambda)$ is usually intractable since we do not know π. To avoid this, we need to relate $\mathbf{G}(q_\lambda)$ to its empirical version,

$$\widehat{\mathbf{G}}\left(q_\lambda\right) \;=\; |\mathbf{D}_{\tau_*}|^{-1} \sum_{(\mathbf{x},y)\in\mathbf{D}_{\tau_*}} \mathbb{E}_{\mathbf{w}\sim q_\lambda}\left[\ell\Big(\eta_{\mathbf{w}}(\mathbf{x}), y\Big)\right], \qquad (16)$$

which replaces the unknown data distribution π with a finite set of training examples $\mathbf{D}_{\tau_*}$. This can then be achieved by applying the classical PAC-Bayes bound of (McAllester, 1999) on $\mathbf{G}(q_\lambda)$ and $\widehat{\mathbf{G}}(q_\lambda)$. The adapted result is formally stated in Theorem 1 below.

Theorem 1 (PAC-Bayes Bound on Prototype Space) *Let π denote τ_*'s data distribution and let $\mathbf{D}_{\tau_*}$ denote a finite collection of a few shots of training examples drawn independently from π. For any $\delta \in (0, 1)$ and learned prior $p_\gamma(\mathbf{w}|\tau_*)$, with probability at least $1 - \delta$ over $\mathbf{D}_{\tau_*} \sim \pi$:*

$$\mathbf{G}(q_\lambda) \;\leq\; \widehat{\mathbf{G}}(q_\lambda) + \sqrt{\frac{\mathbb{KL}\left(q_\lambda \| p_\gamma\right) \;+\; \log\frac{|\mathbf{D}_{\tau_*}|}{\delta}}{2|\mathbf{D}_{\tau_*}| - 1}} \qquad (17)$$

where $\mathbf{G}(q_\lambda)$ and $\widehat{\mathbf{G}}(q_\lambda)$ are defined in Eqs. (15) and (16) while q_λ and p_γ are shorthand for $q_\lambda(\mathbf{w})$ and $p_\gamma(\mathbf{w}|\tau_*)$.

Thus, using Theorem 1, one can adapt $p_\gamma(\mathbf{w}|\tau_*)$ via optimizing the following **adaptation regularizer**,

$$\mathbf{F}_{\mathcal{R}}\left(\gamma, \lambda\right) \;\triangleq\; \widehat{\mathbf{G}}(q_\lambda) + \sqrt{\frac{\mathbb{KL}\left(q_\lambda \| p_\gamma\right) + \log\frac{|\mathbf{D}_{\tau_*}|}{\delta}}{2|\mathbf{D}_{\tau_*}| - 1}} \qquad (18)$$

which essentially learns an adapted version $q_\lambda(\mathbf{w})$ of $p_\gamma(\mathbf{w}|\tau_*)$ that best fits $\mathbf{D}_{\tau_*}$.

Remark. Note that to apply the PAC-Bayes bound here, the prior $p_\gamma(\mathbf{w}|\tau_*)$ must not depend on $\mathbf{D}_{\tau_*}$ and the lost measure must have value between 0 and 1. Both of which are met by our choice of the embedding model in Section 3.1 and loss function $\ell(\eta, y) = 1 - (\eta)_y$. In addition, note also that the adaptation regularizer $\mathbf{F}_{\mathcal{R}}(\gamma, \lambda)$ above can be optimized with respect to both γ and λ to create a bidirectional feedback channel between the embedding and adaptation components so that both can converge to a beneficial parameter state that fits both the black boxes and few-shot training examples of the new task. As such, our approach can be viewed as learning a *data-driven prior* from the black-box models, which is similar in spirit to previous studies in (Ambroladze et al., 2007; Germain et al., 2009).

Algorithm 1: Multi-Task Model Fusion

input : $\mathbf{U}$ – unlabeled dataset
$\mathbf{D}_{\tau_*}$ – few-shot dataset of τ_*
$\mathbf{B}_{\tau_1} \ldots \mathbf{B}_{\tau_p}$ – pre-trained black boxes
$n_\ell, n_e, n_a, \zeta_{\mathcal{I}}$ and ℓ_r – training parameters
output : Learned parameters (θ, γ, α) and (ϕ, ψ)

initialize (θ, γ, α), (ϕ, ψ) and **draw sample**:
$\mathbf{z}_1 \ldots \mathbf{z}_m \sim p(\mathbf{z})$ and $\mathbf{R} \leftarrow (\mathbf{x}_i, \tau_i, \eta_i)_{i=1}^r$ where
$\mathbf{x}_i \sim \mathbf{U}, \tau_i \in \{\tau_1 \ldots \tau_p\}$ and $\eta_i = \mathbf{B}_\tau(\mathbf{x}_i)$
for $\ell \leftarrow 1$ **to** n_ℓ **do**
  maximize $\mathbb{E}\left[\mathbf{L} - \zeta_{\mathcal{I}}\mathbf{F}_{\mathcal{I}}\right]$ over $(\mathbf{x}, \tau, \eta) \sim \mathbf{R}$
  with $\mathbf{L}$ in Eq. (8) and $\mathbf{F}_{\mathcal{I}}$ in Eq. (14)
  for $e \leftarrow$ **to** n_e **do**
    sample $(\mathbf{x}_e, \tau_e, \eta_e) \sim \mathbf{R}$
    $\Delta_\xi \leftarrow \nabla_\xi \Big[\mathbf{L} - \lambda_{\mathcal{I}}\mathbf{F}_{\mathcal{I}}\Big]\Big|_{(\mathbf{x},\tau,\eta)=(\mathbf{x}_e,\tau_e,\eta_e)}$
    $\xi \leftarrow \xi + \ell_r\Delta_\xi$ where $\xi \in \{\theta, \gamma, \alpha, \phi, \psi\}$
  end
  minimize $\mathbf{F}_{\mathcal{R}}$ via Eq. (18)
  for $a \leftarrow$ **to** n_a **do**
    $\lambda \leftarrow \lambda - \ell_r\nabla_\lambda\mathbf{F}_{\mathcal{R}}(\gamma, \lambda)$
  end
  sample $\mathbf{w}_1 \ldots \mathbf{w}_n \sim q_\lambda(\mathbf{w})$
  approximate $G_{\mathbf{w}_i}(\eta|\mathbf{x};\theta,\alpha)$ via Eq. (10)
end
return $\mathbf{B}_{\tau_*}$ – recomposed via Eq. (12)

Fusion Algorithm. Finally, putting these together yield the following fusion algorithm (see Algorithm 1) which

iterates between (1) maximizing $\mathbb{E}[\mathbf{L}(\mathbf{x},\tau,\eta;\theta,\gamma,\alpha,\phi) - \zeta_{\mathcal{I}}\mathbf{F}_{\mathcal{I}}(\mathbf{x},\tau,\eta;\phi,\psi)]$ with respect to $(\theta,\alpha,\phi,\psi)$ while fixing γ; and (2) minimizing $\mathbf{F}_{\mathcal{R}}(\gamma,\lambda)$ with respect to λ while fixing $(\theta,\alpha,\gamma,\phi,\psi)$.

In step (1), the expectation is over $(\mathbf{x},\tau,\eta) \sim \mathbf{R}$. $\mathbf{L}$ and $\mathbf{F}_{\mathcal{I}}$ are optimized simultaneously – the two objectives are traded off via tunable parameter $\zeta_{\mathcal{I}}$. This allows us to segregate generic and specific encoding information in separate but composable latent space, as shown empirically in Section 4.1. Steps (1) and (2) will also be iterated to allow information from minimizing $\mathbf{F}_{\mathcal{R}}$ to propagate through (ϕ,ψ), which in turn guides the decomposition process in step (1).

Likewise, information from maximizing $\mathbf{L} - \zeta_{\mathcal{I}}\mathbf{F}_{\mathcal{I}}$ in step (1) will be forwarded to step (2) to guide the adaptation. This encourages the decomposition to recognize and discard artifact noises (e.g., observational noise that pertains only to a particular task) that are not generalizable to new tasks. This will be shown empirically later in Section 4.1.

4. Experiments

This section presents our empirical results to demonstrate that (1) our method is capable of decoupling task-agnostic and task-specific information of each black-box model (Section 4.1) into separate but composable representations (Section 4.2); and that (2) these representations can be recomposed and adapted into a new model that performs well on a new task with only a few shots of training data (Section 4.3). This is achieved with the following experimental settings:

Task Description. We simulate a multi-task fusion scenario that involves black-boxes learning from the MNIST dataset (LeCun et al., 2010), n-digit MNIST dataset (Oh et al., 2018), and Mini-ImageNet dataset (Ravi & Larochelle, 2017). Given black boxes $\mathbf{B}_{\tau_1}\ldots\mathbf{B}_{\tau_p}$ pre-trained on images of a subset of classes, and a new task τ_* concerning the differentiation between another subset of (potentially) unseen classes, the task is to synthesize a new model $\mathbf{B}_{\tau_*}$ that performs well on τ_* despite its limited data $\mathbf{D}_{\tau_*}$

Comparison Baseline. We compare the classification accuracy of our algorithm's fused model $\mathbf{B}_{\tau_*}$ on the new task against the following baselines: (a) an additive model $\mathbf{B}_{+}$ that sums up and re-normalizes the prediction scores provided by the private black boxes $\{\mathbf{B}_{\tau_i}\}_{i=1}^{p}$; (b) a pointwise-max model $\mathbf{B}_{\max}$ that scores a candidate label by the highest scores given by $\{\mathbf{B}_{\tau_i}\}_{i=1}^{p}$; (c) a modified version of MAML (Finn et al., 2017) adapted to private data setting[1]; and (d) a few-shot model FS trained from scratch using only the few shots of data from the new task. The full parameterization of our model and these baselines are detailed in Appendix B.

[1]In MAML (Finn et al., 2017), one has access to the training data of each model and its parameters, which provides more information than our private and pre-trained setting (see Appendix A).

4.1. Task-Agnostic Embedding

To demonstrate that our method can find a decoupling and task-agnostic representation of a pre-trained model which segregates its generic and specific patterns into separate latent coordinates, we randomly sample a digit image $\mathbf{x}$, a model $\mathbf{B}_\tau \sim \{\mathbf{B}_{\tau_1}\ldots\mathbf{B}_{\tau_p}\}$[2] and extract its corresponding class probabilities vector $\eta = \mathbf{B}_\tau(\mathbf{x})$.

For visualization purpose, this section focuses on the MNIST dataset. First, a digit-concept encoding $\mathbf{z}$ is sampled from $q_\phi(\mathbf{z}|\mathbf{x},\eta)$. Fixing $\mathbf{z}$, we vary the specific pattern $\mathbf{w}$ over its latent space and generate the corresponding reconstructed images (one per latent coordinate $\mathbf{w}$) $\mathbf{x}' \sim p_\theta(\mathbf{x}|\mathbf{w},\mathbf{z})$. The results (see Figs. 2a/b/c/d) interestingly show that all generated images $\mathbf{x}'$ are about the same digit but with different orientations and stroke variations.

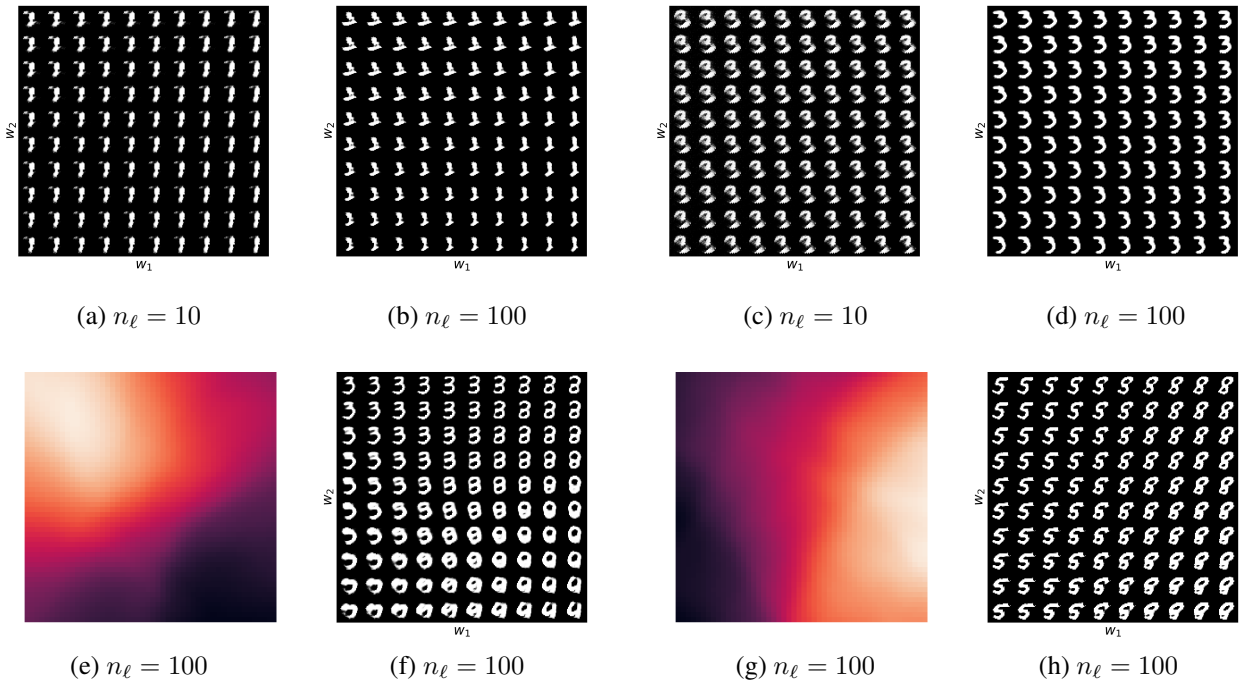
This implies $\mathbf{w}$ and $\mathbf{z}$ are segregated but composable representations that express the black box's prediction at a particular input. Furthermore, it can also be observed from Fig. 2a/b/c/d that the stylistic properties encoded at early stages (i.e., iteration 10) during training are less natural than those encoded at latter stages (i.e., iteration 100). After 10 iterations, there is not much variation in orientation while the strokes are somewhat distorted by artifact noises. After 100 iterations, however, we observe a wider range of orientations (unlike artifact noises) across different digits. This is expected since intuitively, it is much easier to adapt and incorporate such generalizable patterns into the modeling of a new task than using the artifact noises, which will be recognized and discarded by the adaptation regularizer – see Eq. (18) – as we increase the number of training iterations.

4.2. Prototype Visualization

Eq. (10) introduces the notion of a task-agnostic inferential prototype $G_{\mathbf{w}}(\eta|\mathbf{x};\theta,\alpha)$ which captures the black box's predictive behavior around a specific pattern $\mathbf{w}$. To interpret the semantic information captured by these prototypes, we visualize its inferential behaviour by sampling a digit image $\mathbf{x}$, a model $\mathbf{B}_\tau \sim \{\mathbf{B}_{\tau_1}\ldots\mathbf{B}_{\tau_p}\}$ which then allows us to acquire its probabilistic prediction $\eta = \mathbf{B}_\tau(\mathbf{x})$ at $\mathbf{x}$.

A specific pattern $\mathbf{w}$ is then sampled from $q_\phi(\mathbf{w}|\mathbf{x},\tau)$ which results in an inferential prototype $G_{\mathbf{w}}(\eta|\mathbf{x};\theta,\alpha)$. To understand this prototype, we keep $\mathbf{w}$ and η fixed while varying the generic pattern $\mathbf{z}$ over its latent space and generating: (1) the reconstructed image $\mathbf{x}' \sim p_\theta(\mathbf{x}|\mathbf{w},\mathbf{z})$; and (2) a heatmap representing the imposed density of $G_{\mathbf{w}}(\eta|\mathbf{x};\theta,\alpha)$ on the corresponding latent coordinate $\mathbf{z}$. Notice that this is in direct opposite to Section 4.1 wherein we keep $\mathbf{z}$ fix and vary $\mathbf{w}$. The results are plotted in Fig. 2f and Fig. 2h, which show decoded images of multiple digits.

[2]Task τ is selected so that it describes a classification task that involves the true label of $\mathbf{x}$ as a potential output candidate.



(a) $n_\ell = 10$ (b) $n_\ell = 100$ (c) $n_\ell = 10$ (d) $n_\ell = 100$

(e) $n_\ell = 100$ (f) $n_\ell = 100$ (g) $n_\ell = 100$ (h) $n_\ell = 100$

Figure 2. **[TOP – fix z and vary w]** plots of reconstructed images of handwritten digits 1 – *see plots (a) and (b)* – and 3 – *see plots (c) and (d)* – via $p_\theta(\mathbf{x}|\mathbf{w}, \mathbf{z})$ after $n_\ell = 10$ and 100 training iterations with respect to different samples of $\mathbf{w}$ drawn from the encoder $q_\phi(\mathbf{w}|\mathbf{x}, \eta)$ of a sampled image $\mathbf{x}$ of the corresponding digit and outcome distribution $\eta = \mathbf{B}_\tau(\mathbf{x})$ for a sampled task τ that involves digits 1 and 3; and **[BOTTOM – fix w and vary z]** plots of activation heat-maps and encoded digit patterns of two inferential prototypes $G_\mathbf{w}(\eta|\mathbf{x}; \theta, \alpha)$ corresponding to two fixed samples of $\mathbf{w}$. The encoded digit patterns of these prototypes (via their reconstructed images) – *see plots (f) and (h)* – are visualized across the space of $\mathbf{z}$ while the activation heat-maps – *see plots (e) and (g)* – show which of their encoded patterns were activated strongest when the prototypes took images $\mathbf{x}$ of digits 3 and 8, respectively, as input. Here, brighter colors correspond to higher numerical values which indicate stronger activation. See Appendix C for more visualizations and explanations.

Transferability. In contrast to the results in the previous section, a single $\mathbf{w}$ can be combined with different latent concept $\mathbf{z}$ to generate images of different digits. This illustrates that a specific pattern $\mathbf{w}$ tends to encode properties that are not exclusive to any single digit. This corroborates our proposition that the interaction between the embedding and adaptation losses in Algorithm 1 would gear the decomposition towards inferential patterns with transferable information that can generalize well across different digits.

As a concrete example, the reconstructed images in Fig. 2f are generated from the latent $\mathbf{w}$ that underlies $G_\mathbf{w}$ and they all exhibit a two-circle pattern that manifests itself in digits 3 and 8. Likewise, the decoded images from a prototype visualized in Fig. 2h exhibit a S-like stroke pattern that can be seen very commonly in both digits 5 and 8, thus demonstrating their ability to capture transferable knowledge patterns.

Pattern Activation. On the other hand, the corresponding heat-maps over the decoded images in Fig. 2e and Fig. 2g show which part of the prototype's encoded knowledge is activated by an input image. The prototype activation at a latent coordinate $\mathbf{z}$ is computed via $p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{w})p_\alpha(\eta|\mathbf{z})$ following Eq. (10). Fig. 2e and Fig. 2g thus show strong prototype activation at decoded images visualizing digits 3 and 8, which share the same digit-concept with the sampled images used to generate them. Thus, this demonstrates the prototypes' semantic consistency in their pattern activation.

4.3. Multi-Task Model Fusion

Finally, we report the performance of our algorithm in synthesizing and adapting a new model to solve a new task from previous models on three different datasets. In particular, we evaluate and compare the performance of our model against a set of baselines with respect to a series of 2-way classification tasks on MNIST and n-digit MNIST as well as 5-way classification tasks on Mini-ImageNet.

First, we partition the training classes into 2 or 3 sub-datasets $\mathbf{D}_{\tau_1}, \mathbf{D}_{\tau_2}$ and $\mathbf{D}_{\tau_3}$, and train one black-box models $\mathbf{B}_{\tau_1}, \mathbf{B}_{\tau_2}$ and $\mathbf{B}_{\tau_3}$ for each of the corresponding sub-dataset. In Table 1 above, different settings are presented in the left-most column. Each setting is indexed with a dataset name, which is followed by the number of black-box models (either 2 or 3) and a letter S or U that indicates respectively

1-SHOT	$\mathbf{B}_{+}$	$\mathbf{B}_{\text{MAX}}$	$\mathbf{B}_{\tau_*}$ (OURS)	$\mathbf{B}_{\text{MAML}}$	FS
MNIST-2-S	$\mathbf{96.25 \pm 1.06}$	96.25 ± 1.06	$\mathbf{94.25 \pm 4.60}$	92.13 ± 1.60	80.75 ± 13.7
N-MNIST-3-S	99.02 ± 0.71	$\mathbf{99.12 \pm 0.71}$	$\mathbf{96.25 \pm 0.35}$	80.79 ± 2.06	77.11 ± 7.07
MINI-IMAGENET-3-S	87.20 ± 3.75	$\mathbf{87.21 \pm 3.01}$	$\mathbf{87.10 \pm 1.02}$	41.38 ± 2.13	26.45 ± 0.55
MNIST-2-U	50.25 ± 0.35	50.75 ± 1.06	$\mathbf{78.56 \pm 2.70}$	73.92 ± 7.32	76.75 ± 5.30
N-MNIST-3-U	48.11 ± 4.95	48.25 ± 6.01	$\mathbf{94.02 \pm 1.41}$	77.25 ± 7.71	92.50 ± 0.70
MINI-IMAGENET-3-U	21.80 ± 3.78	22.41 ± 2.02	$\mathbf{42.80 \pm 1.11}$	40.78 ± 2.01	26.17 ± 0.78

5-SHOT	$\mathbf{B}_{+}$	$\mathbf{B}_{\text{MAX}}$	$\mathbf{B}_{\tau_*}$ (OURS)	$\mathbf{B}_{\text{MAML}}$	FS
MNIST-2-S	$\mathbf{97.00 \pm 2.12}$	$\mathbf{97.00 \pm 2.12}$	$\mathbf{95.07 \pm 1.41}$	95.49 ± 0.12	83.89 ± 0.16
N-MNIST-3-S	98.91 ± 0.71	98.94 ± 0.71	$\mathbf{99.25 \pm 0.35}$	94.75 ± 0.60	98.33 ± 0.46
MINI-IMAGENET-3-S	87.23 ± 3.60	87.23 ± 3.31	$\mathbf{88.40 \pm 1.02}$	40.96 ± 1.13	27.80 ± 0.15
MNIST-2-U	50.25 ± 0.35	50.75 ± 1.06	$\mathbf{88.77 \pm 0.35}$	87.49 ± 4.83	$\mathbf{89.21 \pm 3.89}$
N-MNIST-3-U	47.51 ± 7.78	47.51 ± 7.78	$\mathbf{99.23 \pm 0.35}$	95.79 ± 0.53	97.77 ± 1.77
MINI-IMAGENET-3-U	21.81 ± 3.61	22.46 ± 2.32	$\mathbf{43.78 \pm 1.97}$	40.77 ± 2.56	27.21 ± 0.34

Table 1. Classification performance of our method and 4 other baselines on three datasets: MNIST (2-way), n-digit MNIST (2-way) and Mini-ImageNet (5-way). Each setting is coded by the dataset name, which is followed by the number of black-box models on related tasks and a letter S or U which indicates whether each test classes were seen or unseen by at least one of the black-box models. In all cases, the reported performance (with mean and standard deviation) is averaged over multiple independent data partitions and runs. The performance of our model $\mathbf{B}_{\tau_*}$ and the best reported performance over all tested methods are highlighted in bold for each experiment setting.

whether each test class was observed by at least one black-box model or all test classes are completely novel and have not been seen by any of the black boxes. The S-setting thus evaluates our algorithm's capability of compiling and aggregating existing knowledge from multiple black-box models to tackle slight variants (with substantial overlapping of classes) of existing tasks, whereas the U-setting tests its ability to further adapt such knowledge to learn and solve novel tasks whose classes have not been seen before.

For each experiment setting, the reported results (with mean and standard deviation) are averaged over multiple independent data partitions and runs. Our reported performance in Table 1 in particular show that:

1. In cases where each test class was observed previously by at least one of the black boxes (e.g., the S-setting which is reported in the top 3 rows), simple heuristic fusion methods such as the additive $\mathbf{B}_{+}$ and pointwise-max $\mathbf{B}_{\max}$ models obtain results that are (a) much better than those of MAML and the FS baseline, especially in the more sophisticated Mini-ImageNet domain; and (b) competitive to ours in most cases (except for MNIST where ours are slightly worse). This (not surprisingly) implies the diminishing gain of meta learning methods over simple aggregation heuristics.

2. Conversely, in cases where all test classes have not been observed previously by any of the black-box models (e.g., the U-setting which is reported in the 3 bottom rows), the performance of the simple heuristic methods decreases significantly and reports accuracies that are not much better than those of a random guess. In contrast, both MAML and our proposed method achieve much better results, which can be observed consistently across all datasets. In all these cases, it is also observed that the performance of our method is much better than MAML's and other heuristic baselines'.

3. Lastly, in most cases (specifically, 5 out of 6 in Table 1), the FS method performs much worse than ours, especially in Mini-ImageNet scenarios. This demonstrates the necessity of knowledge transferability (or meta learning) in multi-task learning, especially in those that have more sophisticated space of output classes (e.g., Mini-ImageNet) whose learning difficulty is further aggravated by the federated nature of data and the limited training data available to train a model from scratch for a new task.

5. Conclusion

This paper introduces a new fusion framework to extract and combine transferable knowledge prototypes from multiple black-box models (solving related tasks) to generate a new model for a new task, which is assumed to be drawn from the same task distribution. We address this challenge by developing a disentangled model embedding which decouples task-specific from task-agnostic information in the latent parameter representation of each black box. This reveals a principled algorithm to distill task-agnostic inferential prototypes via observing how the existing black-box models make predictions at an unlabeled dataset. A prototype distribution can then be learned from the same representation and further adapted to the new task given its training examples. This allows us to recompose the extracted prototypes into a new model that solves the new task effectively.

Acknowledgements

This research/project is supported in part by the Singapore National Research Foundation through the Singapore-MIT Alliance for Research and Technology (SMART) Centre for Future Urban Mobility (FM) and in part by A*STAR under its RIE2020 Advanced Manufacturing and Engineering (AME) Industry Alignment Fund – Pre Positioning (IAF-PP) (Award A19E4a0101). P. Jaillet also acknowledges the research support of the Office of Naval Research (ONR) grant N00014-18-1-2122.

References

Allamraju, R. and Chowdhary, G. Communication efficient decentralized gaussian process fusion for multi-uas path planning. In *American Control Conference*, pp. 4442–4447, 05 2017. doi: 10.23919/ACC.2017.7963639.

Ambroladze, A., Parrado-Hernández, E., and Shawe-taylor, J. S. Tighter pac-bayes bounds. In *Advances in neural information processing systems*, pp. 9–16, 2007.

Chen, J., Low, K. H., Tan, C. K.-Y., Oran, A., Jaillet, P., Dolan, J. M., and Sukhatme, G. S. Decentralized data fusion and active sensing with mobile sensors for modeling and predicting spatiotemporal traffic phenomena. In *Proc. UAI*, pp. 163–173, 2012.

Chen, J., Cao, N., Low, K. H., Ouyang, R., Tan, C. K.-Y., and Jaillet, P. Parallel Gaussian process regression with low-rank covariance matrix approximations. In *Proc. UAI*, pp. 152–161, 2013a.

Chen, J., Low, K. H., and Tan, C. Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. *Robotics: Science and System*, 06 2013b.

Deisenroth, M. P. and Ng, J. W. Distributed Gaussian processes. In *Proc. ICML*, pp. 1481–1490, 2015.

Finlayson, S. G., Bowers, J. D., Ito, J., Zittrain, J. L., Beam, A. L., and Kohane, I. S. Adversarial attacks on medical machine learning. *Science*, 363:1287–1289, 2019.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. ICML*, 2017.

Fu, T., Hoang, T. N., Xiao, C., and Sun, J. Ddl: Deep dictionary learning for predictive phenotyping. In *Proc. IJCAI*, 2019.

Germain, P., Lacasse, A., Laviolette, F., and Marchand, M. Pac-bayesian learning of linear classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 353–360, 2009.

Hoang, Q. M., Hoang, T. N., and Low, K. H. A generalized stochastic variational Bayesian hyperparameter learning framework for sparse spectrum Gaussian process regression. In *Proc. AAAI*, pp. 2007–2014, 2017.

Hoang, Q. M., Hoang, T. N., Low, K. H., and Kingsford, C. Collective model fusion for multiple black-box experts. In *Proc. ICML*, 2019a.

Hoang, T. N. and Low, K. H. A general framework for interacting Bayes-optimally with self-interested agents using arbitrary parametric model and model prior. In *Proc. IJCAI*, pp. 1394–1400, 2013.

Hoang, T. N., Low, K. H., Jaillet, P., and Kankanhalli, M. Nonmyopic ϵ-Bayes-optimal active learning of Gaussian processes. In *Proc. ICML*, pp. 739–747, 2014a.

Hoang, T. N., Low, K. H., Jaillet, P., and Kankanhalli, M. S. Active learning is planning: Non-myopic ϵ-Bayes-optimal active learning of Gaussian processes. In *Proc. ECML-PKDD Nectar Track*, pp. 494–498, 2014b.

Hoang, T. N., Hoang, Q. M., and Low, K. H. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In *Proc. ICML*, pp. 569–578, 2015.

Hoang, T. N., Hoang, Q. M., and Low, K. H. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *Proc. ICML*, pp. 382–391, 2016.

Hoang, T. N., Hoang, Q. M., Ruofei, O., and Low, K. H. Decentralized high-dimensional bayesian optimization with factor graphs. In *Proc. AAAI*, 2018a.

Hoang, T. N., Xiao, Y., Sivakumar, K., Amato, C., and How, J. Near-optimal adversarial policy switching for decentralized asynchronous multi-agent systems. In *Proc. ICRA*, 2018b.

Hoang, T. N., Hoang, Q. M., Low, K. H., and How, J. P. Collective online learning of Gaussian processes in massive multi-agent systems. In *Proc. AAAI*, 2019b.

Hong, S., Xiao, C., Hoang, T. N., Ma, T., Li, H., and Sun, J. Rdpd: Rich data helps poor data via imitation. In *Proc. IJCAI*, 2019.

Huang, K., Xiao, C., Hoang, T. N., and Sun, J. Caster: Predicting drug interactions with chemical substructure representation. In *Proc. AAAI*, pp. 702–709, 2020.

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

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

Low, K. H., Gordon, G. J., Dolan, J. M., and Khosla, P. Adaptive sampling for multi-robot wide-area exploration. In *Proc. IEEE ICRA*, pp. 755–760, 2007.

Low, K. H., Dolan, J. M., and Khosla, P. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *Proc. ICAPS*, pp. 233–240, 2009.

Low, K. H., Yu, J., Chen, J., and Jaillet, P. Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In *Proc. AAAI*, pp. 2821–2827, 2015.

McAllester, D. A. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pp. 164–170, 1999.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proc. AISTATS*, pp. 1273–1282, 2017. URL http://arxiv.org/abs/1602.05629.

Oh, S. J., Murphy, K., Pan, J., Roth, J., Schroff, F., and Gallagher, A. Modeling uncertainty with hedged instance embedding. In *International Conference on Learning Representations (ICLR)*, 2018.

Podnar, G., Dolan, J. M., Low, K. H., and Elfes, A. Telesupervised remote surface water quality sensing. In *Proc. IEEE Aerospace Conference*, 2010.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2017.

Ruofei, O. and Low, K. H. Gaussian process decentralized data fusion meets transfer learning in large-scale distributed cooperative perception. In *Proc. AAAI*, pp. 3876–3883, 2018.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Singh, S. P. and Jaggi, M. Model fusion with optimal transport. 2019. URL https://arxiv.org/abs/1910.05653.

Xiao, C., Hoang, T. N., Hong, S., Ma, T., and Sun, J. Cheers: Rich model helps poor model via knowledge infusion. *IEEE Transactions on Knowledge and Data Discovery*, 2019.

Xu, J. and Wang, F. Federated learning for healthcare informatics, 2019.

Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y., and Ahn, S. Baysian model-agnostic meta-learning. In *Proc. NeurIPS*, 2018.

Yu, H., Hoang, T. N., Low, K. H., and Jaillet, P. Stochastic variational inference for bayesian sparse gaussian process regression. In *Proc. IJCNN*, 2019.

Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, T. N., and Khazaeni, Y. Bayesian nonparametric federated learning of neural networks. In *Proc. ICML*, 2019a.

Yurochkin, M., Argawal, M., Ghosh, S., Greenewald, K., and Hoang, T. N. Statistical model aggregation via parameter matching. In *Proc. NeurIPS*, pp. 10954–10964, 2019b.

Zhang, Y., Hoang, T. N., Low, K. H., and Kankanhalli, M. Information-based multi-fidelity bayesian optimization. In *Proc. NIPS Workshop on Bayesian Optimization*, 2017.

Zhao, P., Liu, S., Chen, P. Y., Hoang, T. N., Xu, K., Kailkhura, B., and Lin, X. On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method. In *Proc. ICCV*, pp. 121–130, 2019.

A. Comparison with Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017)

In this section, we highlight and explain the key differences between the experiment setting of our proposed method and that of MAML (Finn et al., 2017):

First, MAML assumes access to all local datasets and that they can be centralized for processing. This allows MAML to (in the context of our MNIST experiment) sample pairs of different digit images from each local dataset to train a specialized neural network (NNs) that learns a discriminative pattern for each pair of different digits. Such patterns are then aggregated via the gradient adaptation procedure described in Section 2 above. That is, with access to the full dataset, MAML is able to choose the most relevant task samples and their distribution that are most representative for the target task with limited training data. This implicit assumption is, however, not valid in many practical settings with private data that cannot be accessed by the meta learning algorithm.

Second, MAML learns a fast adaptable initializer $\gamma = \gamma_*$ – see Eq. (1) – by imposing the same parameterization and choice of training algorithm across tasks. This appears restrictive in task/data federated settings where pre-trained models solving previous tasks might have different choices of parameterization and training algorithm since they could be generated at different times and under different computing platforms. In contrast, our proposed algorithm is devised to cater specifically to such situations where we only have access to those pre-trained models (potentially with different parameterizations) via their black-box interface from which we can query their output η for each candidate input $\mathbf{x}$. This also implies MAML has access to information with higher quality than ours since MAML has access to the ground-truth y instead of η.

B. Parameterization of Generative and Inference Networks in Figure 1

This section provides detailed information regarding the parameterization of our generative and inference networks in Figure 1. In particular, for the MNIST experiments, we have:

Generative Network:

$p_\theta(\mathbf{x}|\mathbf{w}, \mathbf{z})$ is parameterized as a 3-layer NN with 2 hidden, linear layers and θ being its defining parameters. The input layer accepts batches of $(|\mathbf{w}| + |\mathbf{z}| + |\mathbf{x}|)$-dimensional inputs where $|\mathbf{w}| = 2$ and $|\mathbf{z}| = 5$ are the latent dimensions of the corresponding components that form the factorized embedding space. $|\mathbf{x}| = 28 \times 28 = 784$ which represents the dimension of the flatten digit image of size 28×28. Each hidden layer has 100 hidden neurons, each of which is activated by a ReLU function. Neurons of the last layer is activated by a sigmoid function to guarantee that the pixel values of the reconstructed image $\mathbf{x}$ are from 0 to 1. Additionally, the output of the first hidden layer is passed through a batch normalization layer to avoid the gradient update from being influenced by a few unusually large components.

$p_\alpha(\eta|\mathbf{z})$ is likewise parameterized as a 3-layer NN with 2 hidden, linear layers and α being its defining parameters. Its input layer accepts batches of $|\mathbf{z}|$-dimensional inputs where $|\mathbf{z}| = 5$. Its output layer returns 10-dimensional output η. The rest of the parameterization is similar to $p_\theta(\mathbf{x}|\mathbf{w}, \mathbf{z})$ above.

$p_\gamma(\mathbf{w}|\tau)$ is parameterized as a 3-layer NN with 1 hidden, linear layer comprising of 100 hidden neurons with γ defines its set of parameters. Each hidden neuron is activated by a ReLU unit. The input layer accepts batches of 10-dimensional inputs. The output of the hidden layer will be normalized via a batch normalization layer. The normalized output will then be fed separately to two parallel linear layers comprising of $|\mathbf{w}| = 2$ neurons each. The outputs of these layers represent the mean and diagonal covariance of a multivariate Gaussian that distributes $\mathbf{w}$.

$p(\tau)$ and $p(\mathbf{z})$ are, on the other hand, fixed to be the empirical distribution over a discrete set $\{\tau_1, \tau_2, \ldots, \tau_p\}$ and an isotropic multivariate Gaussian.

For Mini-ImageNet experiments, we adapt the likelihood network $p_\theta(\mathbf{x}|\mathbf{w}, \mathbf{z})$ to a 4-layer transpose convolutional network.

Inference Network:

$q_\phi(\mathbf{z}|\mathbf{x}, \eta)$ is parameterized as a 3-layer NN with 2 hidden, linear layer comprising of 100 hidden neurons. Each hidden neuron is activated by a ReLU unit. The input layer accepts batches of $|\mathbf{x}| + |\eta|$-dimensional inputs where as mentioned above $|\mathbf{x}| = 784$ and $|\eta| = 10$. The output of the hidden layer will be normalized via a batch normalization layer. The normalized output will then be fed separately to two parallel linear layers comprising of $|\mathbf{w}| = 2$ neurons each. The outputs of these layers represent the mean and diagonal covariance of a multivariate Gaussian that distributes $\mathbf{z}$.

$q_\phi(\mathbf{w}|\mathbf{x}, \tau)$ is parameterized similar to $q_\phi(\mathbf{z}|\mathbf{x}, \eta)$. The only difference is that the outputs of its final two parallel, linear

layers represent the mean and diagonal covariance of a multivariate Gaussian that distributes $\mathbf{w}$ with $|\mathbf{w}| = 2$.

For Mini-ImageNet experiments, we adapt the inference network $q_\phi(\mathbf{z}|\mathbf{x}, \eta)$ and $q_\phi(\mathbf{w}|\mathbf{x}, \tau)$ to be 4-layer convolutional networks. Note that while we abuse the notation ϕ to denote the collective set of parameters defining the above two inference networks, these are parameterized separately and do not share weights.

Regularizer Network:

$q_\psi(\eta|\mathbf{w})$ is parameterized as a 3-layer NN with 2 hidden layers and θ being its defining parameters. The input layer accepts batches of $|\mathbf{w}|$-dimensional inputs where $|\mathbf{w}| = 2$. Each hidden layer has 100 hidden neurons, each of which is activated by a ReLU function. Neurons of the last layer is activated by a sigmoid function. Additionally, the output of the first hidden layer is passed through a batch normalization layer to avoid the gradient update from running astray. The final output of this network is $|\eta|$-dimensional where $|\eta| = 10$.

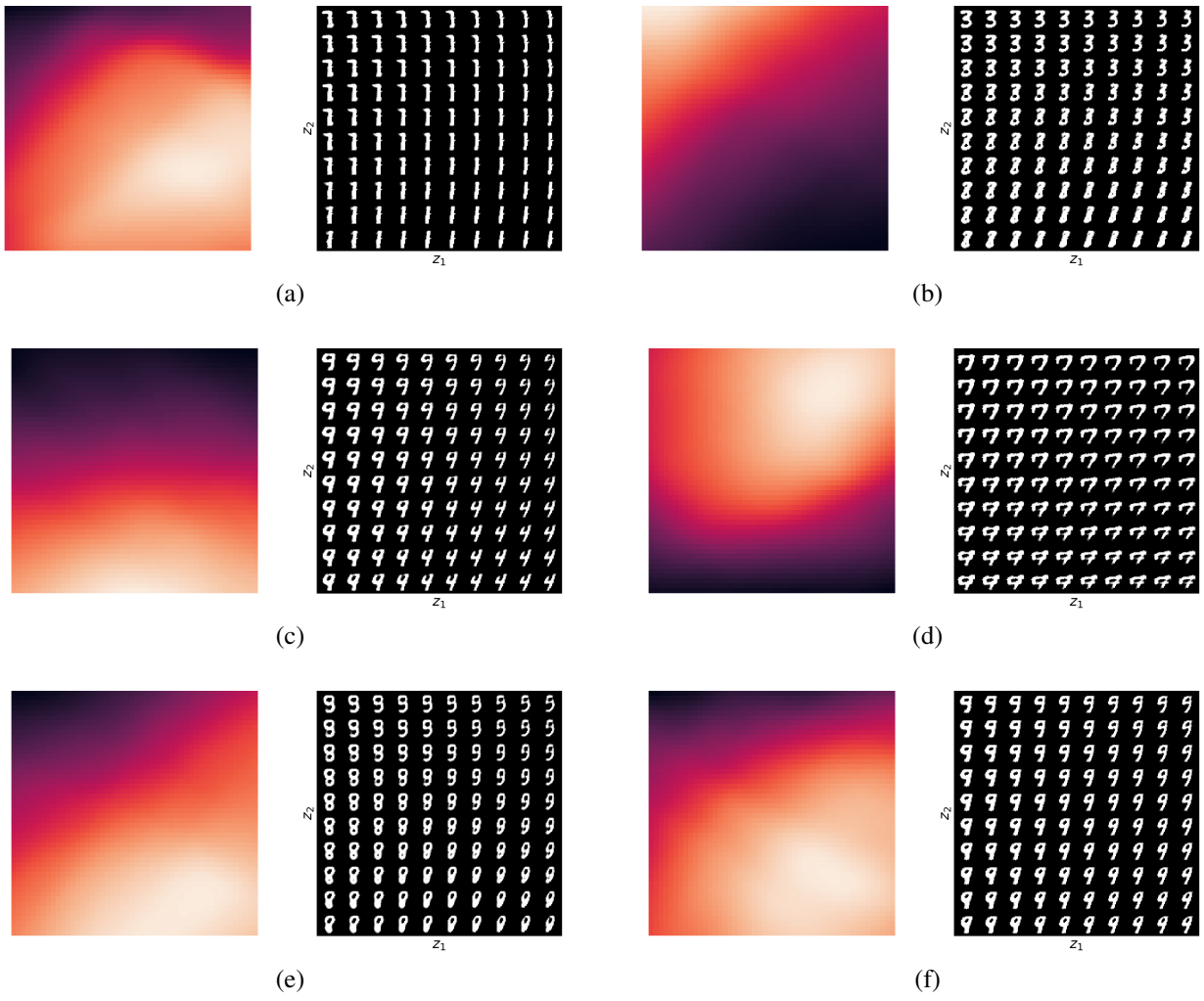


Figure 3. Visualizing plots of 6 different prototypes' activation and encoded patterns with respect to 6 different samples of $\mathbf{w}$. For each prototype, its encoded patterns are visualized via plotting their reconstructed images across the space of $\mathbf{z}$. The corresponding activation heat-maps of these prototypes are also visualized when they process image input of digit (a) 1, (b) 3, (c) 4, (d) 7, (e) 8 and (f) 9. The plotting procedure of these figures was described in detail earlier in Section 4.2.

C. More Explanations for the Experiments in Section 4.2

Figure 2 provides (1) a visualization of handwriting patterns encoded by a particular prototype $G_{\mathbf{w}}(\eta|\mathbf{x};\theta,\alpha)$, which is built around a task-specific pattern $\mathbf{w}$ – see Eq. (10). These patterns are visualized (via the learned decoder $p_\theta(\mathbf{x}|\mathbf{z},\mathbf{w})$) across the latent space of task-agnostic concept $\mathbf{z}$; and (2) the activation (expressed in form of a heat-map where brighter color indicates stronger activation) of these handwriting patterns which is computed via the atomic term $p_\theta(\mathbf{x}|\mathbf{z},\mathbf{w})p_\alpha(\eta|\mathbf{z})$ that forms the prototype's probabilistic prediction in Eq. (10). By inspection, one can further observe that the activation heat-maps of the prototypes shown above are *visually correct* with respect to input images of digits 3 and 8: The activation is strongest at the encoded handwriting patterns corresponding to reconstructed image variants of digits 3 and 8. Interested readers can find more visual excerpts of such prototypes in Figure 3 above.

Finally, to complete our visual demonstration, we further show below how the activation heat-map of the same prototype (i.e., same $\mathbf{w}$ and $\mathbf{z}$-grid) changes when we change its input $\mathbf{x}$ from one digit to another. Again, by inspection, one can observe that both activation heat-maps are *visually correct*: The activation scores are highest at those encoded handwriting patterns that can be reconstructed into similar image variants of the input images. These visual evidences therefore support our claim that the decomposed prototypes are semantically coherent. That is, they always activate the right handwriting patterns (among their encoded patterns) when processing an arbitrary digit image.

Remark. We also note that (as observed in the visual excerpts below) not all handwriting patterns are captured by a single prototype. In fact, each single prototype is shown to only capture a few patterns that only make up a subset of digits. Thus, to perform well in a new classification context that involves unseen digits, these prototypes need to be combined to form the right patterns that characterize the test digits well. This is achieved via solving Eq. (12) above which is the recomposition step that follows the embedding and decomposition procedure detailed in Algorithm 1 above.

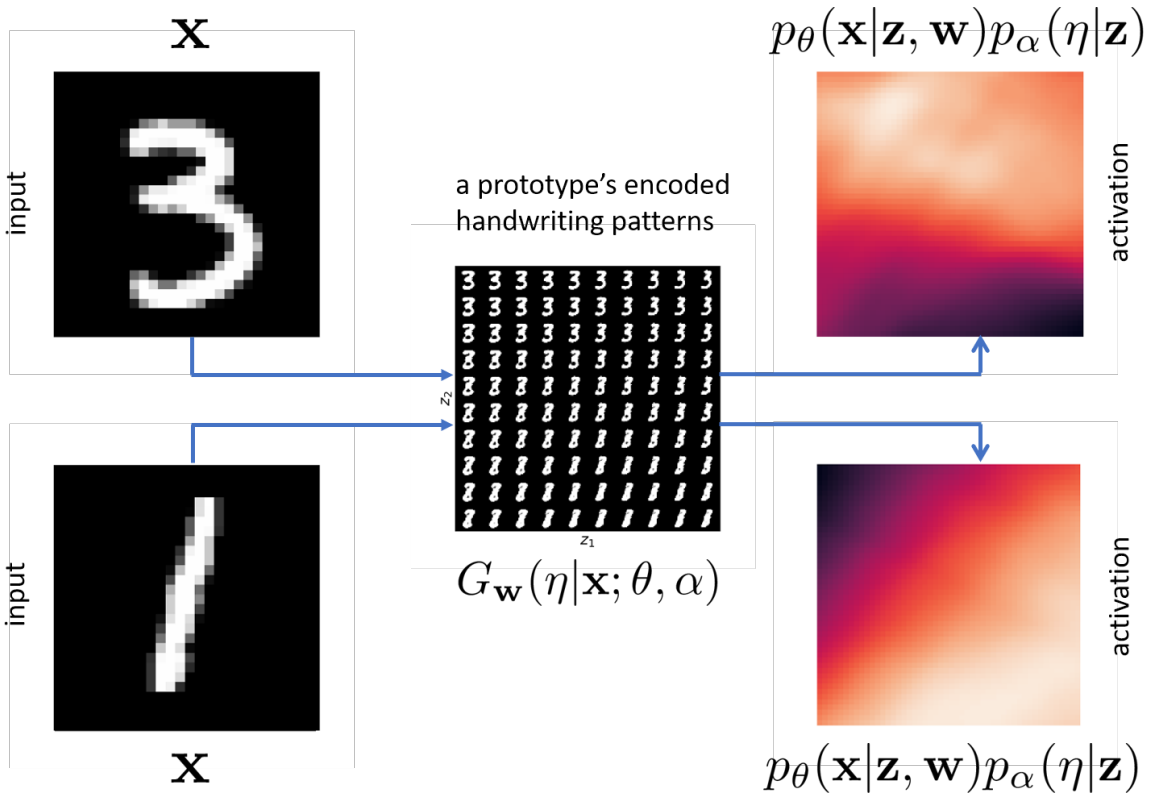


Figure 4. Visualizing plot of activation changes when we change the image input $\mathbf{x}$ to a fixed prototype. When $\mathbf{x}$ is an image of digit 3, the encoded patterns of the prototype that correspond to an image variant (after being decoded) of digit 3 receive strongest activation (i.e., locations annotated with brightest colors). Likewise, for the same prototype, when $\mathbf{x}$ is an image of digit 1, the activation becomes strongest at patterns that can be decoded in a variant image (slightly distorted) of digit 1.