Meta-Amortized Variational Inference and Learning

Kristy Choi [*1] **Mike Wu** [*1] **Noah Goodman** [1 2] **Stefano Ermon** [1]

Abstract

How can we learn to do probabilistic inference in a way that generalizes between models? Amortized variational inference learns for a single model, sharing statistical strength across observations. This benefits scalability and model learning, but does not help with generalization to new models. We propose meta-amortized variational inference, a framework that amortizes the cost of inference over a family of generative models. We apply this approach to deep generative models by introducing the MetaVAE: a variational autoencoder that learns to generalize to new distributions and rapidly solve new unsupervised learning problems using only a small number of target examples. Empirically, we validate the approach by showing that the MetaVAE can: (1) capture relevant sufficient statistics for inference, (2) learn useful representations of data for downstream tasks such as clustering, and (3) perform meta-density estimation on unseen synthetic distributions and out-of-sample Omniglot alphabets.

1. Introduction

A wide variety of problems in modern AI can be posed as probabilistic inference in generative models. While traditional inference techniques solve each inference independently, *amortized inference* (Gershman & Goodman, 2014) aims to solve multiple inferences for a given model together—*learning to do inference* for that model. This approach has been particularly fruitful when applied to variational inference (Jordan et al., 1999; Wainwright et al., 2008; Blei et al., 2017) where amortization across observations solves a serious problem with scaling to large data sets (Rezende et al., 2014; Kingma & Welling, 2013). In this paper we explore amortizing not just over the observations for a single model, but further amortizing the cost of inference over *different generative models*.

More precisely, suppose we have a family of generative models where for each family member, we would like to perform scalable inference. Then, we would ideally design an efficient, amortized inference model that takes as input: (1) a suitable representation of the target probabilistic model, (2) an inference query, and (3) observed data, and outputs an approximation of the desired posterior distribution. We note that this inference model is not intended to be universal, but rather tailored to a specific family where each model is similar in structure. Inspired by meta-learning, we denote this "doubly-amortized" inference problem as *meta-inference* and let a *meta-distribution* refer to the probability distribution over the family of probabilistic models.

The challenge is generalization: we wish to draw correct inferences *efficiently* on unseen distributions that are either sampled from the meta-distribution or "close" to it. This challenge is especially pertinent for latent variable models such as the variational autoencoder (VAE), where the amortized inference network is used to map data points to latent representations. In this work, we introduce the MetaVAE, a VAE that meta-amortizes the inference procedure across a family of generative models. We use the MetaVAE to perform: (1) meta-unsupervised learning, where we leverage the underlying meta-distribution to find good representations on previously unseen distributions for downstream tasks; and (2) meta-density estimation, where we can properly estimate the marginal distribution with very few data points from an unseen target distribution.

2. Preliminaries

2.1. Exact and Approximate Inference

Let $p_{data}(\mathbf{x})$ be an (empirical) data distribution over the observed variables $\mathbf{x} \in \mathcal{X}$. In practice, this is often uniform over a training set D of examples from $\mathcal{X}$. We then define $p(\mathbf{x}, \mathbf{z})$ to be a joint distribution over a set of latent variables $\mathbf{z} \in \mathcal{Z}$ and observed variables $\mathbf{x} \in \mathcal{X}$.

A typical inference query involves computing our posterior beliefs after incorporating evidence into the prior: $p(\mathbf{z}|\mathbf{x}) = p(\mathbf{x}, \mathbf{z})/p(\mathbf{x})$. This quantity is often intractable to compute, as the marginal likelihood $p(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$ requires

[*]Equal contribution [1]Department of Computer Science, Stanford University [2]Department of Psychology, Stanford University. Correspondence to: Kristy Choi and Mike Wu <kechoi@cs.stanford.edu, wumike@stanford.edu>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

integrating/summing over a potentially exponential number of configurations for $\mathbf{z}$.

Instead, we leverage approximate inference techniques such as Markov Chain Monte Carlo (MCMC) sampling (Hastings (1970), Gelfand & Smith (1990)) and variational inference (VI) (Jordan et al. (1999), Wainwright et al. (2008), Blei et al. (2017)) to estimate $p(\mathbf{z}|\mathbf{x})$. In VI, we posit a family of tractable distributions $\mathcal{Q}$ parameterized by ϕ over the latent variables and find the member (called the approximate posterior) $q_{\phi^*} \in \mathcal{Q}$ that minimizes the Kullback-Leibler (KL) divergence between itself and the exact posterior:

$$q_{\phi^*}(\mathbf{z}) = \arg\min_{q_\phi} D_{KL}(q_\phi(\mathbf{z})||p(\mathbf{z}|\mathbf{x}))$$

This $q_{\phi^*}(\mathbf{z})$ then serves as a proxy for the true underlying posterior distribution. We note that the solution will depend on the specific value of the observed (evidence) variables $\boldsymbol{x}$ we are conditioning on. For notational clarity, we rewrite the variational parameters as $\phi_{\boldsymbol{x}}$ to make explicit their dependence on $\boldsymbol{x}$. As noted earlier, one often needs to solve multiple inference queries of the same kind, conditioning on different values of the observed (evidence) variables $\boldsymbol{x}$. The average quality of the variational approximations obtained can be quantified as follows:

$$\mathbb{E}_{\boldsymbol{x}\sim p_{data}(\mathbf{x})}\left[\max_{\{\phi_{\boldsymbol{x}}\}} \mathbb{E}_{q_{\phi_{\boldsymbol{x}}}(\mathbf{z})} \log \frac{p(\mathbf{x},\mathbf{z})}{q_{\phi_{\boldsymbol{x}}}(\mathbf{z})}\right] \tag{1}$$

2.2. Amortized Variational Inference

Massively large training sets $\mathcal{D}$ require yet another layer of efficiency, as the computational cost of VI in Eq. 1 scales linearly with the number of data points $|\mathcal{D}|$. We thus leverage a technique known as *amortization*, in which we amortize the computational cost of the inference procedure by casting the per-sample optimization process in Eq. 1 as a supervised *regression* task. Specifically, rather than solving for an optimal $q^*_{\phi_{\boldsymbol{x}}}(\mathbf{z})$ for every data point $\mathbf{x}$, we learn one deterministic mapping $f_\phi : \mathcal{X} \to \mathcal{Q}$ to *predict* $q^*_{\phi_x}(\mathbf{z})$ as a function of $\boldsymbol{x}$. Often, we choose to concisely represent f_ϕ as a conditional distribution, denoted by $q_\phi(\mathbf{z}|\mathbf{x}) = f_\phi(\mathbf{x})(\mathbf{z})$.

This procedure introduces an *amortization gap*, in which the less flexible parameterization of the inference network results in replacing the original objective as shown in Eqn. 1 with the following lower bound:

$$\max_\phi \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log \frac{p(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \tag{2}$$

This gap refers to the suboptimality caused by amortizing the variational parameters over the entire training set, as opposed to optimizing for each training example individually (pulling the max out of the expectation in Eq. 2). This tradeoff in expressiveness, however, enables significant computational speedups and generalization to new values of the observed variables.

2.3. Latent Variable Models

Of particular importance to latent variable modeling is the variational autoencoder (VAE), a generative model trained to maximize the log marginal likelihood of the data:

$$\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log p(\mathbf{x})] = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[\log \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}\right] \tag{3}$$

as a function of a set of trainable parameters θ.

As an optimization objective, Eqn. 3 is intractable. Instead, we can derive the Evidence Lower Bound (ELBO) to Eqn. 3 using $q_\phi(\mathbf{z}|\mathbf{x})$ as a tractable amortized inference model:

$$\mathbb{E}_{x\sim p_{\text{data}}}[\log p(x)] \geq \mathbb{E}_{x\sim p_{\text{data}}}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]\right] \tag{4}$$

With Eqn. 4 as an objective, we can train the VAE by jointly optimizing ϕ, θ. Post-optimization, the latent variables $\mathbf{z}$ are learned features inferred by $q_\phi(\mathbf{z}|\mathbf{x})$ that can be used in generic unsupervised learning tasks (e.g. clustering).

We may also derive an alternative formulation of the ELBO where denoting $q_\phi(\mathbf{x},\mathbf{z}) = f_\phi(\mathbf{x})(\mathbf{z})p_{\text{data}}(\mathbf{x})$ we get:

$$\begin{aligned}\mathcal{L}(\phi,\theta) &= -D_{\text{KL}}(q_\phi(\mathbf{x},\mathbf{z})\|p_\theta(\mathbf{x},\mathbf{z})) &(5)\\ &= -D_{\text{KL}}(p_{\text{data}}(\mathbf{x})\|p_\theta(\mathbf{x})) \\ &\quad - \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x}))] &(6)\end{aligned}$$

Eqn. 6 comprises a maximum likelihood term with a regularization penalty that encourages the learned model to have posteriors that can be approximated by the amortized inference model (Shu et al., 2018).

3. Meta-Amortized Variational Inference

Recall a (singly)-amortized inference model for $p(\boldsymbol{x},\boldsymbol{z})$

$$\max_\phi \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\mathbf{x})}\left[\mathbb{E}_{f_\phi(\mathbf{x})} \log \frac{p(\mathbf{z})p(\mathbf{x}|\mathbf{z})}{f_\phi(\mathbf{x})(\mathbf{z})}\right]$$

which attempts to approximate $p(\mathbf{z}|\mathbf{x})$ for various choices of $\mathbf{x} \sim p_{data}$. This is the original setting where we consider repeated inference queries from the same model but evaluated on many values of the observed variables $\mathbf{x}$.

Now imagine that we are interested in not one but a set of models, $\mathcal{J}_\mathcal{I} = \{p_{\theta_i}(\boldsymbol{x},\boldsymbol{z}), i \in \mathcal{I}\} = \{p(\boldsymbol{z})p_{\theta_i}(\boldsymbol{x}|\boldsymbol{z}), i \in \mathcal{I}\}$. We assume that the random variables in these models have the same domains $(\mathcal{X},\mathcal{Z})$, but the relationships between the random variables may be different. Further, we make the key simplifying assumption that for each model, we care about the same query $p_\theta(\mathbf{z}|\mathbf{x})$. Finally, we assume to have some knowledge of typical values of the observed variables for each model in $\mathcal{J}_\mathcal{I}$. Formally, we assume to

have a set $\mathcal{M}_{\mathcal{I}} = \{p_i(\mathbf{x}), i \in \mathcal{I}\} \subseteq \mathcal{M}$ of marginal distributions over the observed variables, e.g., a set of data distributions. Here $\mathcal{M}$ denotes the set of all possible marginal distributions over $\mathcal{X}$. Let $p_{\mathcal{M}} : \mathcal{M}_{\mathcal{I}} \to [0, 1]$ denote a distribution over $\mathcal{M}_{\mathcal{I}}$. For example, $p_{\mathcal{M}}$ may be uniform over a finite number of training datasets. As $p_{\mathcal{M}}$ is a distribution over distributions, we refer to it as a *meta-distribution*.

The standard approach to amortize over a set of models is:

$$\mathbb{E}_{p_i \sim p_{\mathcal{M}}} \max_{\phi} \mathbb{E}_{p_i(\mathbf{x})} \left[\mathbb{E}_{f_\phi(\mathbf{x})} \log \frac{p_{\theta_i}(\mathbf{x}, \mathbf{z})}{f_\phi(\mathbf{x})(\mathbf{z})} \right] \tag{7}$$

where we separately fit an amortized inference model for each $p_{\theta_i}(\mathbf{x}, \mathbf{z})$. However, we propose to doubly-amortize the inference procedure as follows:

$$\max_{\phi} \mathbb{E}_{p_i \sim p_{\mathcal{M}}} \mathbb{E}_{p_i(\mathbf{x})} \left[\mathbb{E}_{g_\phi(p_i, \boldsymbol{x})} \log \frac{p_{\theta_i}(\mathbf{x}, \mathbf{z})}{g_\phi(p_i, \boldsymbol{x})(\mathbf{z})} \right] \tag{8}$$

where the original mapping $f_\phi(\mathbf{x})$ is replaced by an amortized mapping $g_\phi(p_i, \mathbf{x})$ that takes the marginal distribution $p_i(\mathbf{x})$ and an observation $\mathbf{x}$ to return a posterior. Formally, we call such a mapping, $g_\phi : \mathcal{M} \times \mathcal{X} \to \mathcal{Q}$, a *meta-inference model*. Given a single inference query, this doubly-amortized inference component must be robust across varying marginals and evidence. The hope is that g_ϕ will generalize over $\mathcal{M}$, and possibly to a larger set of sufficiently similar, but previously *unseen* models.

3.1. Meta-Amortized Variational Learning

Obtaining such a set $\mathcal{J}_{\mathcal{I}} = \{p_{\theta_i}(x, z), i \in \mathcal{I}\}$ of similarly related generative models is difficult. However, just as amortized variational inference works particularly well when learning the parameters of the generative model jointly with those of the amortized inference model, we can "meta-learn" a *set* of generative models jointly with a *single* doubly-amortized inference model.

To meta-learn a VAE, we can jointly optimize the parameters of the meta-inference network ϕ and the parameters of each generative network θ_i, $i \in \mathcal{I}$ according to this objective:

$$\max_{\phi} \mathbb{E}_{p_i \sim p_{\mathcal{M}}} \left[\max_{\theta_i} \mathcal{L}_{\phi, \theta_i}(p_i) \right] \tag{9}$$

where

$$\mathcal{L}_{\phi, \theta_i}(p_i) = -D_{KL}(p_i(\mathbf{x}) g_\phi(p_i, \mathbf{x}) || p(\mathbf{z}) p_{\theta_i}(\mathbf{x}|\mathbf{z})) \tag{10}$$

and $p_i(\mathbf{x}) g_\phi(p_i, \mathbf{x})$ denotes the distribution defined implicitly by $\mathbf{x} \sim p_i(\mathbf{x})$ and $\mathbf{z} \sim g_\phi(p_i, \mathbf{x})$. We denote this lower bound as the MetaELBO, and refer to the VAE with meta-inference as the MetaVAE.

We can rewrite the MetaELBO to a more interpretable form, as in Eqn. 6. Similar to f_ϕ, our doubly-amortized mapping g_ϕ can be represented as a conditional distribution, denoted $q_\phi(\mathbf{z}|\mathbf{x}, p_i) = g_\phi(p_i, \mathbf{x})(\mathbf{z})$. Then,

$$\begin{aligned} \mathcal{L}_{\phi,\theta}(p_i, \mathbf{x}) &= -D_{\mathrm{KL}}(p_i(\mathbf{x}) q_\phi(\mathbf{z}|\mathbf{x}, p_i) || p(\mathbf{z}) p_{\theta_i}(\mathbf{x}|\mathbf{z})) \\ &= -D_{\mathrm{KL}}(p_i(\mathbf{x}) || p_{\theta_i}(\mathbf{x})) \\ &\quad - \mathbb{E}_{\mathbf{x} \sim p_i}[D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}, p_i) || p_{\theta_i}(\mathbf{z}|\mathbf{x}))] \end{aligned}$$

As in (Shu et al., 2018), this MetaELBO has a maximum likelihood term and a regularization term but for each distribution $p_i(\mathbf{x})$, thereby encouraging the meta-amortized inference model to perform well across distributions p_i sampled from the meta-distribution $p_{\mathcal{M}}$.

Finally, we state a property of the MetaELBO: if $|\mathcal{M}| = 1$ and $p_1 \in M = p_{\text{data}}$, then the MetaELBO decomposes to the standard ELBO and $g_\phi(p_i, \mathbf{x}) = f_\phi(\mathbf{x})$.

3.2. Representing the Meta-Inference Model

In Eqn. 10, if we parameterize $g_\phi(p_i, \mathbf{x})$ as a neural network, it is not clear how to represent a distribution, $p_i(\mathbf{x})$ as input. One of the main insights from this work is to "discretize" the marginal distribution as a finite set of samples, $D_i = \{x_j \in p_i(\mathbf{x}) | j = 1, ..., N\}$, or a *dataset*. We can use a dataset, D_i as a surrogate for p_i and define an "empirical" analogue to $g_\phi(p_i, \mathbf{x})$, denoted as $\hat{g}_\phi : \mathcal{X}^N \times \mathcal{X} \to \mathcal{Q}$, which maps a dataset with N samples and an observation to a posterior. Then, there is an equivalent analogue of Eqn. 10 where a marginal, $p_i(\mathbf{x})$ is replaced by a dataset, D_i.

In practice, for some dataset D and $\mathbf{x} \in \mathcal{X}$, we set $\hat{g}_\phi(D, \mathbf{x}) = r_{\phi_2}(\text{CONCAT}(x, h_{\phi_1}(D))$ where $\phi = \{\phi_1, \phi_2\}$, $h(\cdot)$ is a recurrent neural network (RNN) over an arbitrary ordering of the elements in D, and $r(\cdot)$ is a two layer multilayer perceptron (MLP). Each generative model $p_{\theta_i}(\mathbf{x}, \mathbf{z})$, $i \in \mathcal{I}$ is also parameterized by a MLP with identical architecture as $r(\cdot)$. We refer to $h(\cdot)$ as the *summary network* and to $r(\cdot)$ as the *aggregation network*.

3.3. Fully Bayesian VAE

The proposed MetaVAE has an interesting relationship to a fully Bayesian VAE where one would explicitly model a posterior distribution over parameters. More precisely, this involves the factorization of the joint:

$$p(\mathbf{x}, \mathbf{z}, \theta) = p(\mathbf{x}|\mathbf{z}, \theta) p(\mathbf{z}) p(\theta) \tag{11}$$

where $p(\theta)$ is a prior distribution over the parameters. Then, the appropriate inference network would be $q_\phi(\mathbf{z}|\mathbf{x}, \theta)$ i.e. an inference model amortized over a family of generative models $\{p(\mathbf{x}, \mathbf{z}, \theta), \theta \in \Theta\}$. If Θ is a discrete set, then the fully Bayesian VAE is analogous to a MetaVAE.

In practice, the fully Bayesian VAE is difficult to train because Bayesian neural networks are extremely sensitive to hyperparameter choices and initializations. By discretizing Θ to a finite set, we make the optimization problem easier.

3.4. Instantiations of the MetaVAE

The meta-amortized inference procedure is flexible, meaning that it can be instantiated in a variety of ways depending on the probabilistic task. Here we describe two particular instantiations that are used in our experiments. The first

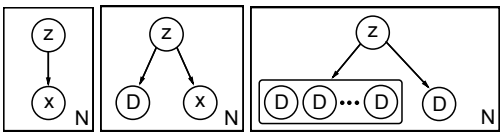


Figure 1. Plate diagrams for (a) VAE, (b) MetaVAE where an observation is a data point, (c) MetaVAE where an observation is a dataset, $D = \{x_i\}$. Let N be the number of observations.

setup (shown in Fig. 1b) is as described in Sec. 3: there exists a meta-inference model $\hat{g}_\phi(D_i, \mathbf{x})$ that takes as input an observation and a dataset. Unless otherwise stated, we default to this instantiation.

An alternative setup (Fig. 1c) imposes an additional layer of abstraction: a single observation is now a dataset $D \in \mathcal{X}^N = \{\mathbf{x}_1, ..., \mathbf{x}_N\} \sim p_i$. The meta-inference model is $\hat{g}_\phi(\{D_i\}, D)$ where $\{D_i\}$ is a sequence of datasets whose elements from drawn from p_i. This $\hat{g}_\phi$ requires a second RNN that ingests a dataset, and returns a single hidden vector. After applying the RNNs, the resulting hidden vector $\mathbf{y} = \text{RNN}(D)$ and sequence of hidden vectors $\{\mathbf{y}_i\} = \text{RNN}(D_i), D_i \in \{D_i\}$ are analogous to Fig. 1b, where we treat a hidden vector as an observed variable.

4. Related work

There exists a rich body of work on meta-learning, particularly in the supervised learning setting with the goal of rapid adaptation to unseen classification tasks (Ravi & Larochelle (2016), Santoro et al. (2016), Vinyals et al. (2016), Snell et al. (2017)). A popular line of work formulates proper initialization as the workhorse of successful meta-learning, such as (**?**, Grant et al. (2018), Yoon et al. (2018)). In many ways, our meta-amortized inference procedure can be thought of as learning a good initialization of for an inference model on a new target distribution.

Meta-learning for unsupervised tasks has also been explored by (Metz et al. (2018)), who learn the weight updates for good representation learning. Several lines of work have tackled the problem of few-shot density estimation, with approaches ranging from attention mechanisms (Rezende et al. (2016)), memory-augmented models (Bornschein et al. (2017)), weight-updates for conditional generative models (Reed et al. (2017)), and hierarchical models (Edwards & Storkey (2016), Hewitt et al. (2018)). Our architecture shares similarities to both the Neural Statistician (Edwards & Storkey (2016)) and the Variational Homoencoder (Hewitt et al. (2018)): we also derive salient features of each dataset with a summary network. Our model's distinguishing factor, then, is on doubly amortizing the inference procedure over a family generative models. To the best of our knowledge, this is a novel contribution.

5. Experimental Results

First, we probe the characteristics and generalization ability of the meta-inference model in two synthetic settings, and then we demonstrate its applicability to meta-density learning using OMNIGLOT.

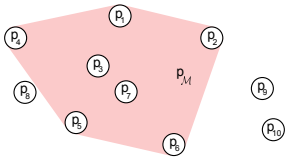


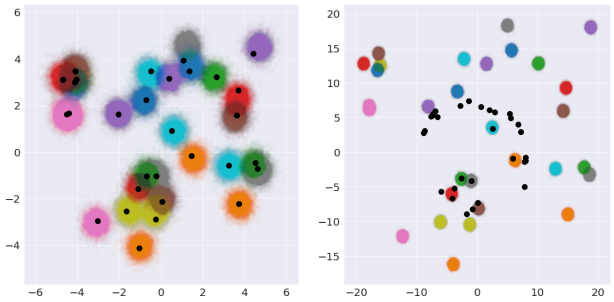
Figure 2. Let $p_1, ..., p_5 \in p_\mathcal{M}$ be distributions used in meta-learning. Weak generalization refers to p_6, p_7; strong generalization refers to p_8, p_9, p_{10}.

5.1. 2D Gaussian Datasets

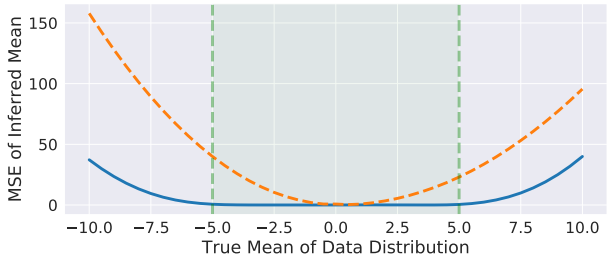
In this experiment, the set of marginals, $\mathcal{M}$, is composed of two-dimensional distributions (e.g., Gaussian) with parameters (e.g., mean and variance) that vary within a fixed range $[\alpha, \beta]$. We amortize over 30 sampled marginals and consequently, optimize 30 generative models. The meta-distribution is uniform over $\mathcal{M}$. Critically, each generative model, $p_{\theta_i}(\mathbf{x}|\mathbf{z})$ is parameter-free ($\theta_i = \emptyset$), thereby encouraging the latent variable $\boldsymbol{z}$ to capture the sufficient statistics of the true distribution, $p_i(\mathbf{x})$. Each generative model $p_{\theta_i}(\mathbf{x}, \mathbf{z})$ is also given the correct distribution family that $p_i(\mathbf{x})$ belongs to. However, the meta-inference model g_ϕ is not given any prior knowledge; it is tasked with matching marginals with the correct families. As sufficient statistics only make sense across a set of observations, we use the second setup of the MetaVAE (Fig. 1c). The measure of success is then how close we can infer the sufficient statistics with no additional training (zero shot) for (1) unseen distributions from $\mathcal{M}$ and (2) unseen distributions outside of $\mathcal{M}$. We refer to (1) as *weak generalization* and (2) as *strong generalization*. We first explore a members of the exponential family one-at-a-time, then proceed to multiple members of the exponential family at the same time.

Gaussian Marginals In this setting, each distribution $p_i \in p_\mathcal{M}$ is Gaussian with a fixed spherical covariance of 0.1 and a mean uniformly sampled from $U(-5, 5)$ i.e. $\alpha = -5$, $\beta = 5$. The summary network and aggregation network have 64 hidden dimensions for all layers. To

measure weak generalization, we sample new means from $U(-5, 5)$ that are previously unseen. For strong generalization, we sample means from $U(-20, 20)$. We find that



(a) Gaussian (Weak) (b) Gaussian (Strong)

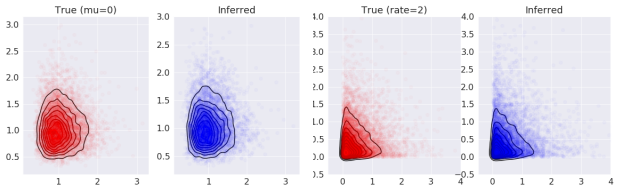


(c) Generalizability Spectrum

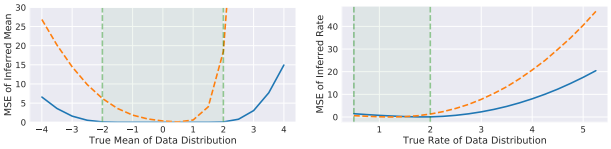
Figure 3. Colored circles represent 30 different $p_i \sim p_{\mathcal{M}}$; black dots represent the inferred means from the meta-inference model. (a) New distributions sampled from $p_{\mathcal{M}}$; (b) New distributions sampled from outside of $p_{\mathcal{M}}$. (c) plots the mean squared error between the true mean and the inferred mean as the true mean of p_i tiles $[-10, 10]$. The green region shows the span of the meta-distribution. The orange line shows a singly-amortized VAE trained on a single p_i with mean $[-1.2, 1.1]$ (randomly chosen).

the MetaVAE is successfully able to learn the means (the only sufficient statistic) of the underlying Gaussians. Interestingly, in Fig. 3a, as you move closer to the boundary of the meta-distribution, the inference quality decreases (see purple Gaussian near $(5, 5)$). In Fig. 3, we can convincingly see that the meta-inference model is almost bounded within the [5,5] square centered at the origin. Finally, from Fig. 3c, we see that doubly-amortizing increases the inference quality dramatically over a singly-amortized model, even for distributions far from $p_{\mathcal{M}}$.

Log Normal and Exponential Marginals Similar to the above setting, we sample 30 log Normal distributions with a fixed spherical covariance of 0.1 and means from $U(-2, 2)$. For strong generalization, we sample from $U(-4, 4)$. We also study the exponential distribution by choosing 30 $p_i(\mathbf{x})$ with a rate sampled from $U(0, 3)$ i.e. $\alpha = 0$, $\beta = 3$. To measure strong generalization, we sample from $U(0, 5)$.



(a) Log Normal (Weak) (b) Exponential (Weak)



(c) Log Normal Spectrum (d) Exponential Spectrum

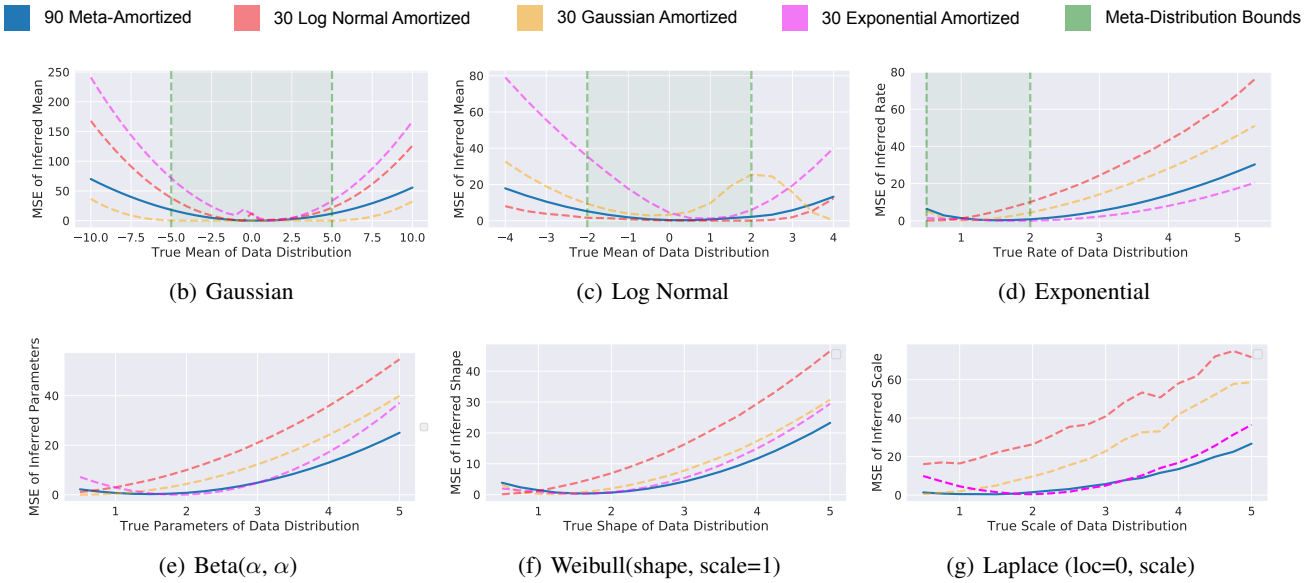
Figure 4. (a) Comparison of samples from the an unseen distribution $p_i \in p_{\mathcal{M}}$ (red) and samples the log normal distribution defined by the inferred sufficient statistic (blue). (b) Similar visualization for exponential distributions as in (a). (c) and (d) show the mean squared error between the true sufficient statistic and the inferred one (mean for log Normal, rate for exponential). The orange line is a non-amortized VAE trained on a single randomly chosen distribution ($[-0.5, 1.8]$ for log Normal; $[1.4, 2.8]$ for exponential).

Many Exponential Families The natural next step is to amortize over many types of exponential families. We sample 30 Gaussian, 30 log Normal, and 30 exponential (with same meta-distributions as above) and train a single meta-inference model. We measure weak and strong generalization as done previously. But, we also measure an even stronger notion of generalization: can we do inference for unseen members of the exponential family?

Fig. 5 compares the performance of our 90 distribution amortized MetaVAE to three different MetaVAEs, each of which is amortized over 30 distributions from a single exponential family. Fig. 5(a-c) show examples of weak generalization. As expected, the best performing model is the MetaVAE amortized on distributions only from that family. However, the 90-amortized MetaVAE only performs slightly worse, beating the remaining two models dramatically. Fig. 5(d-f) show results for 2D distributions over (1) Weibull distributions with a fixed scale of 1, (2) Laplace distributions with a fixed location of 0, and (3) Beta distributions with equal shape parameters. Critically, none of these distributions lie in $p_{\mathcal{M}}$. We find that the 90-amortized MetaVAE consistently outperforms any of the legioned baselines. This suggests that doing inference over the exponential families together enables the model to learn more robust representations.

5.2. 2D Mixtures of Gaussians

Next, we test the MetaVAE's ability to perform clustering and density estimation. A distribution $p_i \in p_{\mathcal{M}}$ is a mix-



(b) Gaussian (c) Log Normal (d) Exponential

(e) Beta(α, α) (f) Weibull(shape, scale=1) (g) Laplace (loc=0, scale)

Figure 5. Comparison of the generalization capabilities of a MetaVAE amortized over three members of the exponential family versus MetaVAE amortized over only a single member. In each subplot, the blue line represents a meta-inference model trained on 30 Gaussian distributions, 30 Exponential distributions, and 30 Log Normal distributions. The other dotted lines are each associated with a meta-inference model trained on 30 of a single type of distribution. Each subplot shows a new unseen distribution drawn either from the meta-distribution (a,b,c) or from another exponential family completely (d,e,f). No additional training was done on the unseen distribution.

ture of Gaussians, where each component is a Gaussian with fixed isotropic covariance $\sigma^2 = 0.1$, and the means are drawn from $U(-5, 5)$. The two Gaussians are mixed equally: $p_i = \frac{1}{2}\mathcal{N}(\mu_1, 0.1) + \frac{1}{2}\mathcal{N}(\mu_2, 0.1)$. We assign each mixture component a label of 0 or 1. We then amortize over $\{10, 30, 50, 100\}$ of such generated mixtures and evaluate whether our meta-inference model can successfully: (1) cluster each mixture component; and (2) estimate the 3 unseen mixture densities, each with means drawn from $U(-5, 5)$, $U(3, 7)$, $U(10, 20)$ respectively.

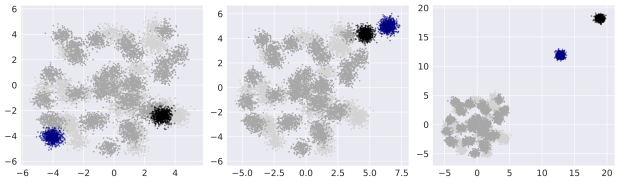


Figure 6. 30 mixtures drawn from the meta-distribution. We plot (in color) 3 unseen distributions whose parameters are drawn from (left) $U(-5, 5)$; (middle) $U(3, 7)$; (right) $U(10, 20)$.

We model $\mathbf{z}$ as a binary latent variable that denotes mixture component membership. We note that the true clustering is exchangeable, only recoverable up to a permutation. We fix the number of hidden units to 10 for all neural networks, and optimize using exact enumeration of the ELBO.

Clustering We first test zero-shot clustering performance on an unseen dataset $p_i \sim p_{\mathcal{M}}$ distributed according to the same underlying meta-distribution. We report average error rates for the zero-shot clustering setup in Table 1. We

samples	10	20	30	50
10 GMMs	0.167	0.212	0.161	0.169
30 GMM	0.167	0.158	0.161	0.169
50 GMM	**0.083**	**0.099**	0.161	0.169
100 GMM	0.084	0.114	0.161	0.169

Table 1. Zero-shot clustering performance on an unseen dataset, $p_i \sim p_{\mathcal{M}}$. We report clustering accuracies over 1000 different sampled datasets, while varying the number of samples used by the RNN to generate a summary statistic at test time.

find that the MetaVAE successfully learns to cluster in the zero-shot case (8%). Amortizing over more meta-datasets seem to help improve the model's clustering performance, although the model trained with 100 mixtures seem to indicate signs of overfitting to the meta-training set.

Next, we extract the pre-trained meta-inference models and train a new generative network on each of 3 unseen data distributions, evaluating the final clustering performance. We only use $\{5, 10, 15, 20\}$% of the test distribution for training. As shown in Figure 4a, the model is able weakly generalize across all levels of meta-training, outperforming the VAE baseline with the exception of the 100 GMM meta-encoder – a phenomena consistent with the results shown in Table 1, i.e., overfitting to the meta-training set. How-

ever, Fig. 4(b,c) shows that meta-training does not seem to provide significant gains in generalization performance on marginals far from $p_{\mathcal{M}}$.

Density Estimation We repeat the same experimental procedure as before, but evaluate final test ELBOS after training the MetaVAEs on small proportions of the unseen data distribution. As expected, Fig. 5a shows that a meta-amortized model is able to perform density estimation well on an instantiation from the same underlying meta-distribution; the performance improvements are very slight for the generalization case to $\mu \sim U(3, 7)$, and fails on completely out-of-distribution samples as evidenced by Fig. 5(b,c).

Intuitively, these results indicate that the meta-encoder successfully learned to take as input a dataset (representative of p_i), identify two clusters, and associate a query datapoint $\mathbf{x}$ to the closest cluster. However, this clustering "algorithm" learned in an unsupervised way by the meta-inference network is imperfect (e.g., inferior to k-means), as it shows signs of overfitting to the training meta-distribution.

5.3. MNIST Clustering and Density Estimation

To further test the MetaVAE, we construct a setup analogous to the mixtures of Gaussians experiment with MNIST (LeCun, 1998). Specifically, we hold out two digit classes for evaluation, and generate datasets comprised of pairs of the remaining digits for training. We select a subset of $\{5, 10, 20\}$ combinations out of a total of 28 (8 choose 2) possibilities to pre-train the meta-amortized model. We then ask the model to: (1) cluster the two digit classes; and (2) perform density estimation on an unseen target dataset. We switch to a continuous variant of the MetaVAE with 40-dimensional latent variables to better model the complexity of the data.

We consider two scenarios for evaluation. For *weak generalization*, the new dataset is still drawn from $p_{\mathcal{M}}$. Concretely, this involves evaluating the MetaVAE on one of the eight remaining combinations (out of 28) that were unseen during training time across all amortized models. For *strong generalization*, we test clustering and density estimation performance on the pair of digits that were held-out for the entirety of the meta-training phase.

Zero-Shot Clustering: We extract the MetaVAE's latent representations ($\mathbf{z}$) of the unseen training data, without additional gradient updates, and train a simple logistic regression model with the true labels (0/1 for each digit class). Intuitively, logistic regression finds the best linear split between two clusters in the latent space; note that for it to perform well, such a linear split must already exist. In this sense, the clustering is "zero-shot". To measure performance, we obtain the corresponding latent vectors for the test set and predict the true labels.

For weak generalization, Fig. 9a & b shows the clustering results for two levels of difficulty: digits 1/6 (easy) and 4/9 (hard). For the former, an amortized MetaVAE outperforms the VAE trained on the *full* dataset of 1's and 6's; however, there does not seem to be much benefit in additional amortization (i.e. amortizing over 5 pairs performs as well as 20 pairs). For the more difficult task, adding more combinations improves clustering performance, and the MetaVAE outperforms a VAE trained on half of the target data. For strong generalization, Fig. 9c shows that the meta-inference network is able to obtain less than 2% clustering error *without adapting its encoder parameters to the unseen data distribution*. Further, it outperforms a VAE which has been trained on 100% of the target dataset of 3's and 7's. We note that the meta-inference model has inferred useful representations that allows for good zero-shot clustering performance on a new, unseen dataset.

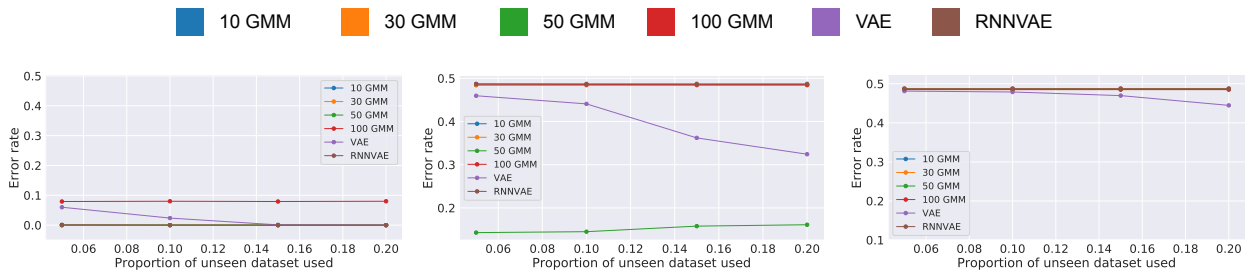
Density Estimation: For strong generalization, we extract the pre-trained meta-inference network and train a new generative model using $\{1, 5, 10, 20\}$% of the target dataset. Fig. 9d shows that we reach a much better test log-likelihood across the board compared to a VAE trained from scratch, and Fig. 9 (e-g) show faster rates of learning for a MetaVAE amortized over 20 combinations of digit pairs as compared to those of a vanilla VAE.

5.4. OMNIGLOT Transfer Density Estimation

To showcase meta-amortized inference beyond synthetic settings, we consider the challenge of transfer density estimation in OMNIGLOT[1]. We reserve the first 25 OMNIGLOT alphabets for a *pre-training phase* where we optimize a MetaVAE amortized over K linear combinations of the 25 alphabets. Precisely, we sample K combination vectors that each sum to one from a Dirichlet distribution with 25 categories. Each combination vector specifies the probability that we sample uniformly form that particular alphabet. Intuitively, we can think of the meta-distribution as constructing K new alphabets. For each new alphabet, we sample 5000 times to create a dataset $D_i, i = 1, ..., K$. The purpose of generating new alphabets in this manner is to expand the training dataset size as some of the original OMNIGLOT alphabets are extremely small and cannot possibly yield good density estimation. The MetaVAE is then amortized across $|\mathcal{M}| = K$ distributions. In practice, $K = 50$.

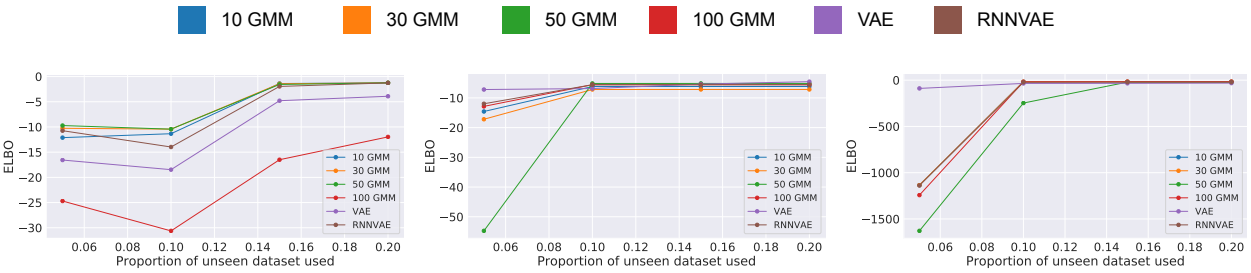
For transfer learning, we consider (case 1) an unseen distribution by sampling a new combination vector over the 25 alphabets used in pre-training, and (case 2) an unseen distribution by sampling a new combination vector over the remaining 35 alphabets *not used in pre-training*. For an unseen distribution, p_i, we initialize the inference network

[1]We use the pre-processed version from (Burda et al., 2015): https://github.com/yburda/iwae/blob/master/datasets/OMNIGLOT/chardata.mat



(b) Weak generalization: $\mu \sim U(-5, 5)$ (c) Generalization: $\mu \sim U(3, 7)$ (d) Strong generalization: $\mu \sim U(10, 20)$

Figure 7. Each sub-figure shows the final clustering performance after training on {5,10,15,20}% of the unseen data distribution. In (a), meta-training on 10, 30, and 50 datasets allows for perfect clustering, on par with the RNNVAE baseline and outperforming the VAE. The 100 GMM meta-trained model overfits. In (b), only the 50 GMM meta-trained model has successfully learned to cluster. In (c), the meta-clustering algorithm fails to generalize to an extremely out-of-sample distribution.



(b) Weak generalization: $\mu \sim U(-5, 5)$ (c) Generalization: $\mu \sim U(3, 7)$ (d) Strong generalization: $\mu \sim U(10, 20)$

Figure 8. Each sub-figure shows the final test ELBOs after training on {5,10,15,20}% of the unseen data distribution. In (a), meta-training on 10, 30, and 50 datasets allows for good generalization performance on an unseen dataset drawn from the same underlying meta-distribution as compared to baselines trained from scratch. In (b), we are able to perform reasonably well in a slightly out-of-sample dataset. In (c), we cannot generalize to datasets that are extremely out-of-distribution.

using the pre-trained meta-inference model and initialize its generative network from scratch. We allow ourselves to train to completion but limit the number of examples from p_i to a small number (making this few-shot). In our experiments, we limit to 100 or 10 examples.

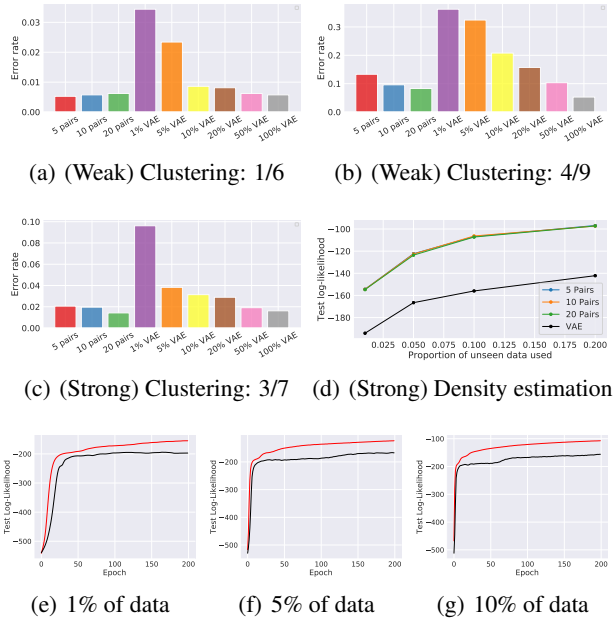
Fig. 10 shows nine experiments, the first three being in case 1, the second three in case 2 and the last three in case 2 but limited to 10 training examples. As expected, meta-amortizing reaches a higher log marginal (and faster) on new distributions in $p_{\mathcal{M}}$. Moreover, we observe worse performance in Fig. 10d as there are little guarantees for distributions outside of $p_{\mathcal{M}}$. What is surprising however, is that meta-amortizing reaches a higher log marginal for the other two distributions composed of entirely new alphabets. This implies that the initialization must be useful in other alphabets, as most hieroglyphs share similarities in design.

Finally, if we restrict how many examples the model can use to transfer to 10, we find that meta-amortizing is still effective (although less so). A MetaVAE and a vanilla VAE (trained on only the 10 examples) converge to the same final log likelihood but the amortized model reaches the optimum faster. Interestingly though, p_4, which did not benefit from meta-amortization under 100 examples, now does just so, suggesting a relationship between relying on prior (amortized) knowledge and observed data.

6. Discussion

Confronted with the theoretical hardness results for exact and approximate inference, the idea of learning approximate inference strategies that are not fully general but tailored to "typical" models of the world is appealing. Our meta-amortization method is particularly useful when we have a set of models with shared structure, and we want to leverage that structure for good few-shot generalization performance on a related target task. We now mention a few observations:

The interesting generalization behavior is in-between weak and strong generalization. With a large enough $|\mathcal{M}|$, we find that meta-amortization leads to weak generalization for distributions in $p_{\mathcal{M}}$. This is sensible as the training distributions build a convex hull that can span $p_{\mathcal{M}}$. In most cases, we cannot hope to expect strong generalization as distributions outside of $p_{\mathcal{M}}$ can be wildly different.

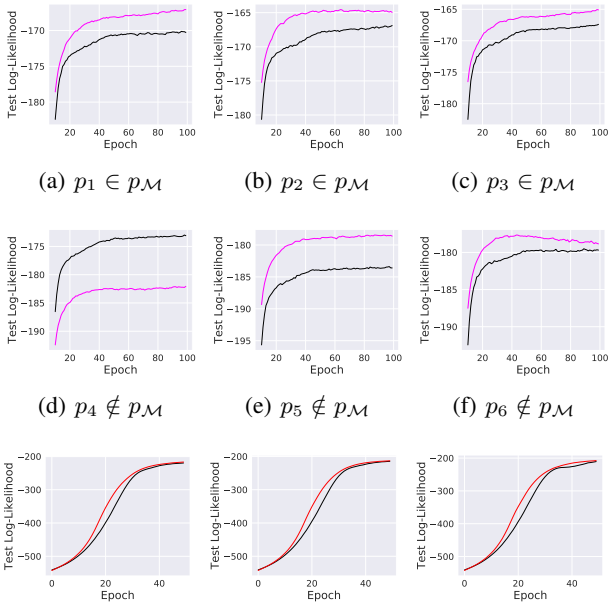


(a) (Weak) Clustering: 1/6 (b) (Weak) Clustering: 4/9

(c) (Strong) Clustering: 3/7 (d) (Strong) Density estimation

(e) 1% of data (f) 5% of data (g) 10% of data

Figure 9. Clustering and density estimation on MNIST. We train a MetaVAE amortized over $\{5, 10, 20\}$ pairs of digit classes and evaluate their performance on unseen pairs from $p_{\mathcal{M}}$ (weak) and outside of $p_{\mathcal{M}}$ (strong). The first row shows that the MetaVAE achieves higher zero-shot clustering performance as compared to a VAE trained on 100% and 50% of the target distribution, still within the meta-distribution. The second row shows that the MetaVAE outperforms a VAE trained on *100% of the out-of-sample target distribution (not in $p_{\mathcal{M}}$)*, and outperforms a VAE across the board for few-shot density estimation. The last row compares test log-likelihoods of the MetaVAE amortized over 20 combinations against those of a vanilla VAE. Colored lines denote MetaVAE models; black lines denote vanilla VAE models trained on the target data.



(a) $p_1 \in p_{\mathcal{M}}$ (b) $p_2 \in p_{\mathcal{M}}$ (c) $p_3 \in p_{\mathcal{M}}$

(d) $p_4 \notin p_{\mathcal{M}}$ (e) $p_5 \notin p_{\mathcal{M}}$ (f) $p_6 \notin p_{\mathcal{M}}$

(g) p_4 (10 examples) (h) p_5 (10 examples) (i) p_6 (10 examples)

Figure 10. Transfer density estimation on OMNIGLOT. We train a MetaVAE amortized over 50 mixtures of 25 OMNIGLOT alphabets. This figure shows test log likelihoods from training independently on three unseen distributions $(p_1, p_2, p_3) \in p_{\mathcal{M}}$ and three unseen mixtures of held-out OMNIGLOT alphabets (p_4, p_5, p_6). For each distribution $p_{1:6}$, we only let the model train on 100 examples (and only 10 examples for the last row in the figure). We train for 100 epochs and 50 epochs for experiments with 100 and 10 examples respectively. Subfigures (a-f) are plotted from epoch 10. Pink and red lines denote MetaVAE models; black lines denote vanilla VAE models trained on only the unseen distribution.

It is the distributions near but outside the edge of $p_{\mathcal{M}}$ (e.g. p_8 in Fig. 2) at which our method demonstrates promise.

Meta-amortization is subject to meta-overfitting. We observe a form of overfitting unique to doubly-amortizing inference, which we call *meta-overfitting*. Meta-overfitting can occur in two ways. In the first scenario, we fail to sufficiently cover the space of $p_{\mathcal{M}}$ by amortizing over *too few datasets*. When this happens, the meta-inference model essentially fails to learn the algorithm of interest (e.g. density estimation) and cannot generalize to even other distributions in $p_{\mathcal{M}}$. In the second scenario, the meta-inference model is trained with too many marginal distributions such that with its limited number of parameters, it fails to capture the correct marginal for any single generative model. Intuitively, it has overfit completely to the underlying meta-distribution. This is exemplified by the 100 GMM-pretrained MetaVAE's inability to cluster in the 2D Gaussian mixture experiments.

7. Conclusion

We introduce meta-amortized variational inference and learning. As far as we know, this is the first instance of amortizing over families of generative models. We find appealing results on density estimation and representation learning, where meta-training leads to improved sample complexity, and hope to explore meta-inference for zero-shot compilation of probabilistic programs.

References

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Bornschein, J., Mnih, A., Zoran, D., and Rezende, D. J. Variational memory addressing in generative models. In *Advances in Neural Information Processing Systems*, pp. 3920–3929, 2017.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Edwards, H. and Storkey, A. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

Gelfand, A. E. and Smith, A. F. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

Gershman, S. and Goodman, N. Amortized inference in probabilistic reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36, 2014.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

Hastings, W. K. Monte carlo sampling methods using markov chains and their applications. 1970.

Hewitt, L. B., Nye, M. I., Gane, A., Jaakkola, T., and Tenenbaum, J. B. The variational homoencoder: Learning to learn high capacity generative models from few examples. *arXiv preprint arXiv:1807.08919*, 2018.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Metz, L., Maheswaranathan, N., Cheung, B., and Sohl-Dickstein, J. Learning unsupervised learning rules. *arXiv preprint arXiv:1804.00222*, 2018.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. 2016.

Reed, S., Chen, Y., Paine, T., Oord, A. v. d., Eslami, S., Rezende, D., Vinyals, O., and de Freitas, N. Few-shot autoregressive density estimation: Towards learning to learn distributions. *arXiv preprint arXiv:1710.10304*, 2017.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Rezende, D. J., Mohamed, S., Danihelka, I., Gregor, K., and Wierstra, D. One-shot generalization in deep generative models. *arXiv preprint arXiv:1603.05106*, 2016.

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

Shu, R., Bui, H. H., Zhao, S., Kochenderfer, M. J., and Ermon, S. Amortized inference regularization. *arXiv preprint arXiv:1805.08913*, 2018.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pp. 3630–3638, 2016.

Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y., and Ahn, S. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pp. 7343–7353, 2018.