Meta-learning richer priors for VAEs

Marcello Negri and **Vincent Fortuin**[*]
ETH Zürich
Jan Stühmer[*]
Samsung AI Centre

Abstract

Variational auto-encoders have proven to capture complicated data distributions and useful latent representations, while advances in meta-learning have made it possible to extract prior knowledge from data. We incorporate these two approaches and propose a novel flexible prior, namely the Pseudo-inputs prior, to obtain a richer latent space. We train VAEs using the Model-Agnostic Meta-Learning (MAML) algorithm and show that it achieves comparable reconstruction performance with standard training. However, we show that this MAML-VAE model learns richer latent representations, which we evaluate in terms of unsupervised few-shot classification as a downstream task. Moreover, we show that our proposed Pseudo-inputs prior outperforms baseline priors, including the VampPrior, in both models, while also encouraging high-level representations through its pseudo-inputs.

1. Introduction

Variational auto-encoders (VAEs) (Kingma and Welling, 2014; Rezende et al., 2014) are deep generative models that allow to learn complicated data distributions. However, VAEs often make use of the isotropic Gaussian prior that may over-regularize the posterior (Hoffman and Johnson, 2016), failing to capture its multi-modal nature. For this reason, several alternative priors have been proposed, such as the Mixture of Gaussians (Dilokthanakul et al., 2016; Nalisnick et al., 2016; Kopf et al., 2021) or the VampPrior (Tomczak and Welling, 2018). We propose the Pseudo-inputs prior to provide more flexibility than the VampPrior while encouraging higher-level representations.

In parallel, high-level representations of the data can also be achieved through meta-learning, which aims at learning to generalise across tasks so that the model can quickly adapt to novel ones. This requires learning specific representations that avoid to re-train in the face of new data (Santoro et al., 2016). We combine VAEs with meta-learning with the aim of generating better latent representations. Specifically, we employ the Model-Agnostic Meta-Learning algorithm (MAML) (Finn et al., 2017) to train VAEs, namely MAML-VAE.

The contributions of this work are three-fold:

- We show VAEs trained with MAML achieve comparable performances with standard training on the full dataset, and better ones on Omniglot with convolutional layers.
- We propose the Pseudo-inputs prior and show it outperforms the standard prior, the Mixture of Gaussians, and the VampPrior on VAEs and on the meta-learnt MAML-VAE model, while being more flexible and capturing higher-level representations.
- We show that meta-learning allows to generate richer latent representations, in terms of higher accuracy in unsupervised few-shot classification, evaluated as a downstream task. Again, the Pseudo-inputs prior achieves the highest accuracy.

[*] equal contribution

2. Methods

2.1. MAML-VAE

We use the standard implementation of VAEs, for which the variational posterior is assumed to be a diagonal Gaussian distribution $q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \mathrm{diag}(\sigma^2_\phi(x)))$ modelled with a neural network parametrised by ϕ. The decoder, on the other hand, is defined as a Bernoulli distribution modelled by a neural network $p_\theta(x|z)$ parametrised by θ. Lastly, we indicate the prior as $p_\lambda(z)$ parametrised by λ. The objective optimised in VAEs is the evidence lower bound (ELBO), which is computed via a Monte Carlo estimate as

$$\mathrm{ELBO}(x; \theta, \phi, \lambda) \simeq \frac{1}{L} \sum_{l=1}^{L} \big[\log p_\theta(x|z^{(l)}_\phi) + \log p_\lambda(z^{(l)}_\phi) - \log q_\phi(z^{(l)}_\phi|x)\big], \tag{1}$$

where $z^{(l)}_\phi \sim q_\phi(z|x)$ using the reparametrization trick (Kingma and Welling, 2014).

In this work, we apply meta-learning to VAE priors by leveraging the MAML algorithm, since its model-agnostic nature allows to maintain the Bayesian structure of the VAE. In short, the idea behind the MAML algorithm is to find the optimal parameters of the model so that in a fixed (small) number of steps over the support set $\mathcal{D}^{supp}$ they can be adapted to unseen tasks over the query set $\mathcal{D}^{query}$. The sought optimal parameters should thus be *sensitive* enough, allowing with small changes to obtain significant improvements in the loss function. The training and test procedures for the resulting model, which we call MAML-VAE, are outlined in Appendix A in Algorithm 1 and 2. In contrast to VAEs, where parameters are updated over an arbitrary number of gradient steps, the MAML-VAE model allows only a single adaptation step over the support set to adapt to the query set, which makes the task harder. It has also been suggested that the optimal parameters found by the MAML algorithm coincide with the optimal parameters for continual learning (Gupta et al., 2020). Intuitively, this should lead to richer high-level latent representations.

2.2. The Pseudo-inputs prior

We propose the *Pseudo-inputs prior* with the intention of providing a more flexible encoding than the VampPrior, while also learning a high-level representation of the dataset. The Pseudo-inputs prior encodes a mixture of Gaussians by mapping K learnable h-dimensional pseudo-inputs $\{u_k\}_{k=1}^K$ to the mean vectors $\{\mu_k\}_{k=1}^K$ in the d-dimensional latent space through a flexible encoding network $f_\psi(\cdot) : u_k \mapsto \mu_k$. The k-th component of the mixture is thus simply obtained by feed-forwarding the k-th pseudo input. We verified experimentally that learning the covariance vectors separately leads to better results. Overall, the probability density encoded by the Pseudo-inputs prior is given by

$$p_\lambda(z) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}\big(z; f_\psi(u_k), \mathrm{diag}(\sigma^2_k)\big), \tag{2}$$

where $\lambda = \{\psi, \{u_k\}_{k=1}^K, \{\sigma^2_k\}_{k=1}^K\}$ are the parameters of the prior. An illustration of the proposed Pseudo-inputs prior is provided in Figure 1.

This encoding allows to freely define the complexity of the network as well as the dimension of the pseudo-inputs, proportionally to how informative they are intended to be. We

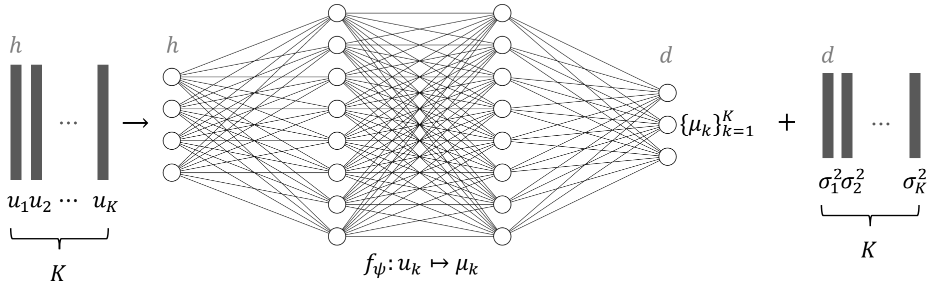


Figure 1: Structure of the proposed Pseudo-inputs prior.

could not learn this high-level representation if we instead encoded the mean vectors and covariances directly. In particular, the Pseudo-inputs prior is meant to learn a particular parametrisation of the network that allows the pseudo-inputs to adapt to different tasks, making it suitable for meta-learning as well as transfer and continual learning. We also suggest that pseudo-inputs could be sampled to alleviate the mode-seeking behaviour of the Kullback-Leibler divergence (Burda et al., 2016), as outlined in Appendix A.

3. Experiments

3.1. Pseudo-inputs prior: rate-distortion analysis

In order to understand the behavior of the Pseudo-inputs prior, we perform the rate-distortion analysis (Alemi et al., 2018) and compare it against the standard prior, the Mixture of Gaussians prior (MoG)—encoding mean and covariance vectors directly—and the VampPrior. In order to do so, we leverage the β-VAE objective proposed in Higgins et al. (2017), which is defined as $\mathcal{L} \coloneqq \mathrm{RE} + \beta \mathrm{KL}$ with the distortion being the reconstruction error $\mathrm{RE} \coloneqq - \int q(x) dx \int q_\phi(z|x) \log p_\theta(x|z) dz$ and the rate being the Kullback-Leibler divergence $\mathrm{KL} \coloneqq \int q(x) dx \int q_\phi(z|x) \log \frac{q_\phi(x|z)}{p_\lambda(z)} dz$. In practice, we trained the VAE model with the β-VAE objective for $\beta \in \{0.01, 0.1, 0.5, 1, 2, 5, 10\}$. We compare the priors in terms of their ELBO ($\mathrm{RE} + \mathrm{KL}$) on Omniglot (Lake et al., 2015) and Quickdraw (Jongejan et al., 2016), which we report in Figure 2 and in Appendix B in Figure 4, respectively.

It is apparent that the least expressive prior is almost everywhere the standard Gaussian, as it achieves the highest ELBOs for most β values. In contrast, the proposed Pseudo-inputs prior achieves the lowest ELBO values for $\beta = 0.5, 1, 2$ consistently for Omniglot and for Quickdraw. This β range is actually the most significant as it is the closest to the actual ELBO bound for $\beta = 1$. In fact, only $\beta = 1$ constitutes a lower bound on the marginal log-likelihood while the more extreme values $\beta = 0.01, 0.1, 5, 10$ allow only to test the auto-encoding and auto-decoding behavior. Furthermore, when $\beta > 1$, the over-pruning biases of variational inference are enhanced (Stühmer et al., 2019). In particular, for $\beta = 1$, the proposed Pseudo-inputs prior significantly outperforms the other priors. Therefore, we can state that based on the rate-distortion analysis, the proposed Pseudo-inputs prior achieves better performances, consistently for Omniglot and Quickdraw, suggesting it is more expressive than the VampPrior, the MoG prior, and the standard prior.

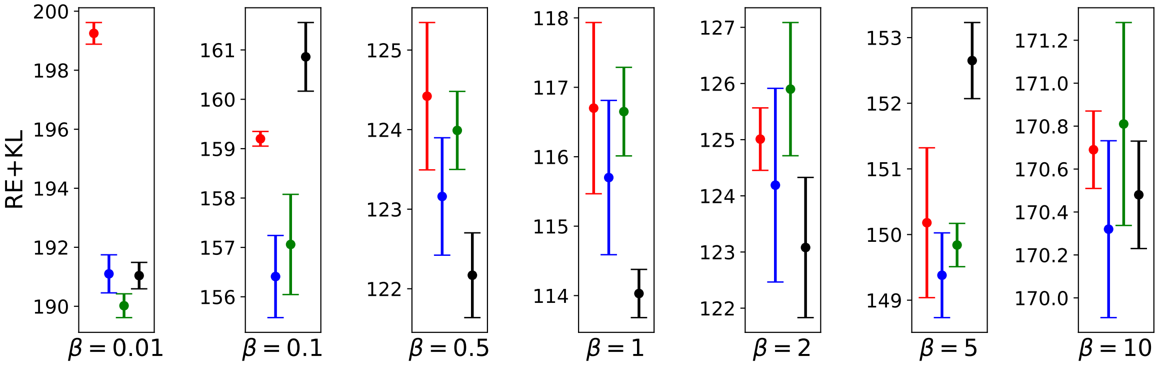


Figure 2: ELBO for VAE trained over β-VAE on Omniglot with standard prior (red), MoG prior (blue), VampPrior (green), and Pseudo-inputs prior (black). Uncertainties are computed as 95% confidence interval over 3 random seeds (10,100,1000).

3.2. MAML-VAE and VAE comparison

In this section, we compare performances between the VAE model and the proposed MAML-VAE model. Furthermore, we show the proposed Pseudo-inputs prior outperforms baseline priors in both models on both Omniglot and Quickdraw. We provide details about how Omniglot and Quickdraw are organised into tasks for meta-learning and about the models in Appendix A. We also compare the models implemented with convolutional layers, namely the MAML-CVAE and CVAE models, to see whether convolutions help generate a latent space suitable for unsupervised few-shot classification and, if so, in what proportion to the MAML-VAE and VAE models. Whenever the "MAML-" prefix is used, all parameters (including the prior, if applicable) are learnt through the MAML algorithm, while standard training is used otherwise. The results obtained are reported in Table 1.

On the one hand, results clearly show the proposed Pseudo-inputs prior outperforms baseline priors. This is true on Omniglot and Quickdraw and consistently across all four models. This suggests that results are robust and that the Pseudo-inputs prior is suitable both for standard and meta-learning, since it is able to efficiently learn a dataset-specific representation and a higher-level one. On the other hand, results show that the VAE and the MAML-VAE models, and their convolutional counterparts CVAE and MAML-CVAE, achieve comparable performances across different priors on Omniglot and Quickdraw. The difference is more evident when convolutional layers are employed as the MAML-CVAE model outperforms the CVAE model over all priors on Omniglot while it achieves worse results on Quickdraw. This is notable, as meta-learning is in general a harder task.

3.3. Unsupervised few-shot classification in the latent space

In order to quantify whether the latent space learnt in the MAML-VAE model is richer than the one learnt with VAEs, we measure its suitability to unsupervised few-shot classification, which we evaluate as a downstream task of prototypical networks (Snell et al., 2017). Specifically, we exploit prototypical networks to evaluate the embedding generated

	Omniglot		Quickdraw	
prior	MAML-VAE	VAE	MAML-VAE	VAE
standard	108.57 ± 1.13	108.20 ± 0.03	187.51 ± 0.19	186.56 ± 0.14
MoG	108.00 ± 0.44	107.37 ± 0.24	185.84 ± 0.16	183.24 ± 0.17
VampPrior	107.07 ± 0.61	106.99 ± 0.01	185.29 ± 0.46	183.19 ± 0.15
Pseudo-inputs	$\mathbf{106.30 \pm 0.08}$	$\mathbf{106.44 \pm 0.08}$	$\mathbf{184.01 \pm 0.30}$	$\mathbf{181.16 \pm 0.05}$
prior	MAML-CVAE	CVAE	MAML-CVAE	CVAE
standard	93.07 ± 0.90	95.59 ± 0.36	160.49 ± 0.18	159.82 ± 0.60
MoG	92.70 ± 0.21	93.95 ± 0.32	156.23 ± 0.48	154.09 ± 0.34
VampPrior	92.15 ± 3.46	98.16 ± 0.89	155.66 ± 1.29	158.90 ± 0.63
Pseudo-inputs	$\mathbf{87.73 \pm 0.07}$	$\mathbf{93.37 \pm 0.21}$	$\mathbf{152.85 \pm 0.36}$	$\mathbf{150.02 \pm 0.40}$

Table 1: Negative marginal test log-likelihood for MAML-VAE and VAE models (fully connected) and for MAML-CVAE and CVAE models (convolutional). Uncertainties are computed as 68% confidence interval over 3 random seeds (10,100,1000). Bold indicates best performing prior.

by variational posterior $q_{\phi^*}(z|x)$ depending on whether it was learnt through the VAE or the MAML-VAE model. Note that the variational posterior is trained through either standard training or the MAML algorithm and that the learnt parameters are left unchanged. This makes the task unsupervised since the models, and hence the variational posteriors, are trained without any label information. Instead of averaging over samples of the variational posterior, we evaluate the latent space generated by its mean vector $\mu_{\phi^*}(x)$ directly. Let us now consider a few-shot task as an N-way classification with S support and Q query shots, taken respectively from $\mathcal{D}^{supp}$ and $\mathcal{D}^{query}$, both drawn from the test set $\mathcal{D}^{test}$. At test time, the support set allows to obtain $\{c_k^{supp}\}_{k=1}^N$ prototypical representations of the N classes as:

$$c_k^{supp} = \frac{1}{|\mathcal{D}_k|} \sum_{(x_i, y_i) \in \mathcal{D}_k^{supp}} \mu_{\phi^*}(x_i), \tag{3}$$

where $\mathcal{D}_k$ is the set of support points labelled with class k. The accuracy of the unsupervised few-shot classification task is then computed on the query set based on the prototypical class representations obtained on the support set. This is computed as softmax over the Euclidean distance between the class prototypes $\{c_k^{supp}\}_{k=1}^N$ and the encoded query points.

We evaluated the models on Omniglot over various N-way S-shot classification tasks: results for the MAML-VAE and VAE model are reported in Table 2 while for the MAML-CVAE and CVAE models in Appendix B in Table 4. On the one hand, we see that the MAML-VAE model achieves significantly higher accuracy than the VAE model across all priors, which indicates that meta-learning encourages a significantly richer latent space. When convolutional layers are employed, the difference shrinks, which indicates that increased capacity of the convolutional encoder and decoder allows for a less expressive prior distribution. However, when either the VampPrior or the Pseudo-inputs prior are employed, it is still beneficial to exploit both contributions. On the other hand, we showed that the

$(N;\ S, Q)$		$(5;\ 1, 19)$	$(5;\ 5, 15)$	$(20;\ 1, 19)$	$(20;\ 5, 15)$
std	MAML-VAE	$\mathbf{57.08 \pm 0.66}$	$\mathbf{75.86 \pm 0.54}$	$\mathbf{33.72 \pm 0.26}$	$\mathbf{52.86 \pm 0.30}$
	VAE	49.23 ± 0.60	69.43 ± 0.54	28.91 ± 0.35	47.76 ± 0.28
MoG	MAML-VAE	$\mathbf{59.74 \pm 0.69}$	$\mathbf{76.50 \pm 0.52}$	$\mathbf{35.17 \pm 0.29}$	$\mathbf{53.63 \pm 0.30}$
	VAE	51.95 ± 0.61	71.41 ± 0.54	30.63 ± 0.25	49.54 ± 0.28
Vamp	MAML-VAE	$\boxed{\mathbf{61.04 \pm 0.66}}$	$\mathbf{78.74 \pm 0.52}$	$\boxed{\mathbf{37.90 \pm 0.30}}$	$\mathbf{57.86 \pm 0.29}$
	VAE	52.18 ± 0.60	70.52 ± 0.59	30.31 ± 0.26	49.41 ± 0.28
Pseudo	MAML-VAE	$\boxed{\mathbf{61.05 \pm 0.65}}$	$\boxed{\mathbf{79.96 \pm 0.49}}$	$\boxed{\mathbf{37.88 \pm 0.31}}$	$\boxed{\mathbf{59.23 \pm 0.28}}$
	VAE	52.24 ± 0.62	72.17 ± 0.49	30.49 ± 0.24	49.83 ± 0.28

Table 2: Accuracy in unsupervised few-shot classification as a downstream task of prototypical networks on Omniglot between the MAML-VAE and VAE models. Accuracy is evaluated over N-way classification with S support images and Q query images. Uncertainties are computed as 95% confidence interval over 1000 tasks. Bold indicates best performing training algorithm while boxes indicate overall best method.

proposed Pseudo-inputs prior achieves the highest accuracy when compared to the other priors across all models and even more so when convolutional layers are employed. Notably, we even achieve comparable results with the unsupervised meta-learning approach CACTUs (Hsu et al., 2019), as detailed in Appendix B in Table 5.

4. Related Work

In recent years, many different priors have been proposed for VAEs (e.g., Fortuin et al., 2019a, 2020; Manduchi et al., 2019, 2021; Jazbec et al., 2020, 2021; Ashman et al., 2020). However, to the best of our knowledge, none of these approaches used meta-learning to improve the priors using related tasks. The prior most related to our approach is the VampPrior (Tomczak and Welling, 2018), which we also use as a baseline in our experiments.

While meta-learning (Thrun and Pratt, 1998; Baxter, 2000) has also been recently connected to Bayesian models (e.g., Grant et al., 2018; Finn et al., 2018; Yoon et al., 2018; Fortuin et al., 2019b; Rothfuss et al., 2021), it has not yet been used for VAEs.

5. Conclusion

We have shown that meta-learning, and specifically the MAML algorithm, can be beneficial for VAEs, since it enables richer latent representations. In particular, we have shown that the MAML-VAE model achieves comparable performances with VAEs in terms of test marginal log-likelihood, despite meta-learning being a much harder task, while it yields better performances in downstream few-shot classification tasks. Moreover, we have proposed the Pseudo-inputs prior, which allows to flexibly encode a mixture of Gaussians while also learning a high-level representation of the data through its pseudo-inputs. We have shown that this prior outperforms the standard prior, the Mixture of Gaussians, and the VampPrior, in standard supervised and meta-learning settings, and on different data sets.

References

Alexander A. Alemi, Ben Poole, Ian Fischer, Joshua V. Dillon, Rif A. Saurous, and Kevin Murphy. Fixing a broken elbo, 2018.

Matthew Ashman, Jonathan So, William Tebbutt, Vincent Fortuin, Michael Pearce, and Richard E Turner. Sparse Gaussian Process Variational Autoencoders. *arXiv preprint arXiv:2010.10177*, 2020.

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

David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer, 2018.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space, 2016.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders, 2016.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus), 2016.

Tristan Deleu, Tobias Würfl, Mandana Samiei, Joseph Paul Cohen, and Yoshua Bengio. Torchmeta: A Meta-Learning library for PyTorch, 2019.

Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning, 2017.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference, 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9537–9548, 2018.

Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. SOM-VAE: Interpretable Discrete Representation Learning on Time Series. In *International Conference on Learning Representations*, 2019a.

Vincent Fortuin, Heiko Strathmann, and Gunnar Rätsch. Meta-Learning Mean Functions for Gaussian Processes. *arXiv preprint arXiv: 1901.08098*, 2019b.

Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. GP-VAE: Deep Probabilistic Time Series Imputation. In *International Conference on Artificial Intelligence and Statistics*, pages 1651–1661. PMLR, 2020.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting Gradient-Based Meta-Learning as Hierarchical Bayes. In *International Conference on Learning Representations*, 2018.

Gunshi Gupta, Karmesh Yadav, and Liam Paull. La-maml: Look-ahead meta learning for continual learning, 2020.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Matthew D. Hoffman and Matthew J. Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound, 2016.

Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning, 2019.

Metod Jazbec, Michael Pearce, and Vincent Fortuin. Factorized Gaussian Process Variational Autoencoders. *arXiv preprint arXiv:2011.07255*, 2020.

Metod Jazbec, Matthew Ashman, Vincent Fortuin, Michael Pearce, Stephan Mandt, and Gunnar Rätsch. Scalable Gaussian Process Variational Autoencoders. In *International Conference on Artificial Intelligence and Statistics*, 2021.

J. Jongejan, H. Rowley, T. Kawashima, J. Kim, and N. Fox-Gieg. The quick, draw! - a.i. experiment. 2016.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes, 2014.

Andreas Kopf, Vincent Fortuin, Vignesh Ram Somnath, and Manfred Claassen. Mixture-of-Experts Variational Autoencoder for clustering and generating from similarity-based representations. *PLoS Computational Biology*, 2021.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

J. MacQueen. Some methods for classification and analysis of multivariate observations. 1967.

Laura Manduchi, Matthias Hüser, Julia Vogt, Gunnar Rätsch, and Vincent Fortuin. DPSOM: Deep probabilistic clustering with self-organizing maps. *arXiv preprint arXiv:1910.01590*, 2019.

Laura Manduchi, Matthias Hüser, Martin Faltys, Julia Vogt, Gunnar Rätsch, and Vincent Fortuin. T-DPSOM: An Interpretable Clustering Method for Unsupervised Learning of Patient Health States. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 236–245, 2021.

Eric Nalisnick, Lars Hertel, and Padhraic Smyth. Approximate inference for deep latent gaussian mixtures. In *NIPS Workshop on Bayesian Deep Learning*, volume 2, page 131, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR.

Jonas Rothfuss, Vincent Fortuin, Martin Josifoski, and Andreas Krause. PACOH: Bayes-Optimal Meta-Learning with PAC-Guarantees. In *International Conference on Machine Learning*, 2021.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks, 2016.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning, 2017.

Jan Stühmer, Richard Turner, and Sebastian Nowozin. ISA-VAE: Independent subspace analysis with variational autoencoders, 2019.

Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pages 3–17. Springer, 1998.

Jakub M. Tomczak and Max Welling. Vae with a vampprior, 2018.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7343–7353, 2018.

Appendix A. Dataset and models

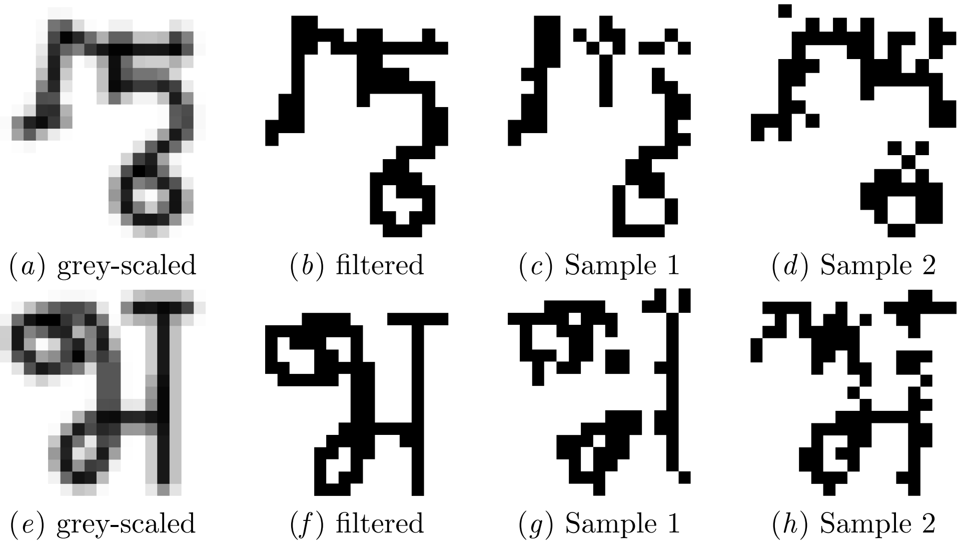
Omniglot Omniglot (Lake et al., 2015) contains images of handwritten characters from 50 alphabets: 30 are intended for training (*background* set) and 20 for test (*evaluation* set). Each character is drawn 20 times from 20 different people. For both the VAE and MAML-VAE models we refer to the background-evaluation split. However, the MAML algorithm requires the dataset to be organised into tasks, which we do in *N-way K-shot* fashion, where training is performed over K support points for a task that consists of N classes. Specifically, we identify a task as learning a particular character, so each task is defined by the 20 handwritten images of each character, which are split into support and query images. Overall, the dataset so defined is composed of 1020 tasks for training, 170 for validation and 420 for test, for a total of 20400, 3400 and 8400 data points, respectively. For the Omniglot meta-learning dataset we leverage the Torchmeta implementation (Deleu et al., 2019), which allows to easily build *N-way K-shot* tasks. In order to ensure images were normalised identically, we used the Torchmeta package for the VAE model as well.

Quickdraw Quickdraw (Jongejan et al., 2016) is a dataset containing 50 million drawings belonging to 345 classes obtained by asking different users to draw the object represented by each class. Since the objects can be drawn in several ways, examples belonging to the same class may look significantly different. The meta-learning task on Quickdraw is thus to capture the "idea" of the represented object, which is a much more challenging task than learning a specific character as in Omniglot. To our knowledge there is no easy-to-use meta-learning implementation of the Quickdraw data set, so we decided to build our own. We selected 200 drawings for each class and we split training, validation and test so that each contains disjoint sets of classes. We picked 50 classes for training, 10 for validation and 25 for test, for a total of 10000, 2000 and 5000 data points, respectively. We report the classes used in each split in Table 3. When choosing the classes for training, validation and test we paid attention to keep a consistent variety of shapes.

Table 3: Classes used in each split of our reduced Quickdraw dataset: 50 classes for training, 10 for validation and 25 for test. Each class contains 200 data points.

train	**validation**	**test**
airplane, ambulance, angel, ant, anvil, apple, arm, asparagus, axe, backpack, banana, bandage, barn, baseball, basket, basketball, bat, bathtub, beach, bear, beard, bed, bee, belt, bench, bicycle, binoculars, bird, blackberry, blueberry, book, boomerang, bottlecap, bowtie, bracelet, brain, bread, bridge, broccoli, broom, bucket, bulldozer, bus, bush, butterfly, cactus, cake, calculator, calendar, camel	*crayon, crocodile, crown, cup, diamond, dishwasher, dog, dolphin, donut, door*	*camera, camouflage, campfire, candle, cannon, canoe, car, carrot, castle, cat, cello, chair, chandelier, church, circle, clarinet, clock, cloud, compass, computer, cookie, cooler, couch, cow, crab*

Both Quickdraw and Omniglot consist of square grey images composed of 28×28 pixels. In this work we decided to binarise each image by applying a filter, namely a mask for which pixels greater than 0.5 are set to 1 while those that are smaller than 0.5 are set to 0. This way a sharper and more continuous image is obtained. In contrast, it is also common to binarise the images through a Bernoulli sampling (Tomczak and Welling, 2018). However, this technique does not guarantee that the resulting shape is continuous, which is a crucial aspect both for the Omniglot and the Quickdraw dataset. In fact, in both cases the images are created through a (mostly) continuous stroke of pen. Examples of how the Bernoulli sampling can distort the resulting images and a comparison with our filtering approach are reported across Figure 3(a)-3(h).


(a) grey-scaled (b) filtered (c) Sample 1 (d) Sample 2

(e) grey-scaled (f) filtered (g) Sample 1 (h) Sample 2

Figure 3: Two grey-scaled Omniglot characters (a) and (e) are binarised. Filtered images preserve the continuity of the pen stroke as in (b) and (f). Bernoulli sampling produces highly different and discontinuous samples as in (c)/(d) and (g)/(h).

Sampling the pseudo-inputs The encoder f_ψ is a mapping from the pseudo-inputs space to the latent space. Instead of learning the pseudo-inputs, they may also be defined over some simple distribution so that its samples are then drawn and mapped in the latent space. One way to do so is to sample the pseudo-inputs from the standard Gaussian distribution $u_k \sim \mathcal{N}(c_k, \sigma^2 1_d)$, where σ^2 is an hyper-parameter and c_k a fixed (or trainable) mean vector, and to do so at each forward pass. Intuitively, the parameter σ^2 controls the variance of the encoded samples $\hat{\mu}_k$, which are obtained as $\hat{\mu}_k = f_\psi(c_k + \sigma\epsilon)$ with $\epsilon \sim \mathcal{N}(0, 1_d)$. Assuming the optimisation converges, samples of the pseudo-inputs are then mapped into samples of the encoded prior distribution. Sampling the pseudo-inputs at each forward pass during optimisation does not guarantee convergence though. However, if the analogous problem where pseudo-inputs are trainable converges, then for sufficiently small ϵ it also converges for trainable c_k values. We show in our experiments that c_k need not to be trainable for convergence and we suggest that sampling pseudo-inputs allows to alleviate the mode-seeking behaviour typical of the Kullback-Leibler divergence (Burda et al., 2016).

Algorithm 1: MAML-VAE: meta-training

```
Input: $\mathcal{D}^{train}$
Output: $\Phi$
Initialise $\Phi := \{\theta, \phi, \lambda\}$;
while not converged do
    Sample a mini-batch of task(s) $\tau$ from $\mathcal{D}^{supp}, \mathcal{D}^{query} \sim \mathcal{D}^{train}$;
    for each task $\tau \in \mathcal{D}^{supp}$ do
        $\Phi' \leftarrow \Phi - \eta_{in} \nabla_{\Phi} \mathcal{L}_{\tau}(\Phi)$;
    end
    $\Phi \leftarrow \Phi - \eta_{out} \nabla_{\Phi} \sum_{\tau \sim \mathcal{D}^{query}} \mathcal{L}_{\tau}\big(\Phi'(\Phi)\big)$;
end
```

Algorithm 2: MAML-VAE: meta-test

```
Input: $\Phi^*, \mathcal{D}^{test}$
Output: $L$
$L \leftarrow 0$;
for $\mathcal{D}^{supp}, \mathcal{D}^{query}$ in $\mathcal{D}^{test}$ do
    $\Phi' \leftarrow \Phi^* - \eta_{in} \nabla_{\Phi} \mathcal{L}_{\tau}(\Phi^*)$;
    $L \leftarrow L + \sum_{\tau \sim \mathcal{D}^{query}} \mathcal{L}_{\tau}(\Phi'(\Phi^*))$;
end
$L \leftarrow L / |\mathcal{D}^{test}|$
```

Details about the VAE and MAML-VAE models The VAE model is trained with a learning rate of 0.0005 whereas the MAML-VAE model with an inner learning rate $\eta_{in} = 0.05$ and an outer learning rate $\eta_{out} = 0.0005$. The MAML-VAE model is trained in a 1-way fashion with 5 support and 15 query shots (20 images per task). We employ ADAM (Kingma and Ba, 2017) as optimizer for the VAE model and for the MAML-VAE model in the outer-loop, while in the inner loop we use simple stochastic gradient descent. The computational bottleneck of the MAML-VAE model is in the outer-loop, which requires back-propagating through the gradients of the inner-loop. Therefore, in MAML-VAE we take a single gradient update in the inner loop. In each model we perform 100 warm-up steps (Bowman et al., 2016) to ensure that the reconstruction error and the Kullback-Leibler term in the ELBO are properly optimised. Furthermore, the stop criterion used for training is in all settings a 50 look-ahead steps early stopping. As suggested in (Tomczak and Welling, 2018), we use a 40-dimensional latent space for both the VAE and MAML-VAE models. Both models are implemented on top of their code, which relies on PyTorch (Paszke et al., 2019). We also provide an implementation of the models that exploits convolutional layers, which we call the CVAE and MAML-CVAE models. In order to make a fair comparison we maintain the same architecture, only replacing fully connected layers with convolutional ones. The trained models are evaluated in terms of the marginal log-likelihood on the test set, which is computed using the importance weighting estimate proposed in the IWAE model (Burda et al., 2016) with 5000 samples. Concerning the Pseudo-inputs prior, its encoder f_{ψ} is implemented with three fully connected layers of dimension $h \times 100$, 100×100, and $100 \times d$

with $h = 50$ and $d = 40$. We applied the Exponential Linear Unit (ELU) (Clevert et al., 2016) as activation function on each layer. In order to make a fair comparison, the MoG prior, the VampPrior and the Pseudo-inputs prior are all implemented with 100 components.

Appendix B. Further results

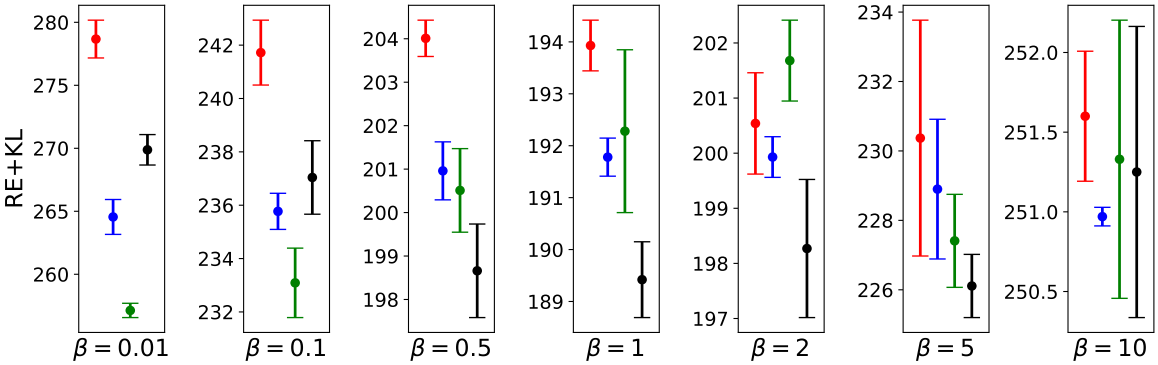


Figure 4: ELBO for VAE trained over β-VAE on Quickdraw with standard prior (red), MoG prior (blue), VampPrior (green), and Pseudo-inputs prior (black). Uncertainties are computed as 95% confidence interval over 3 random seeds (10,100,1000).

$(N;\ S, Q)$		$(5;\ 1, 19)$	$(5;\ 5, 15)$	$(20;\ 1, 19)$	$(20;\ 5, 15)$
std	MAML-CVAE	57.56 ± 0.63	77.22 ± 0.51	39.21 ± 0.29	59.47 ± 0.30
	CVAE	$\mathbf{58.62 \pm 0.64}$	$\mathbf{78.19 \pm 0.51}$	$\mathbf{40.47 \pm 0.28}$	$\mathbf{61.16 \pm 0.29}$
MoG	MAML-CVAE	59.12 ± 0.66	78.89 ± 0.51	39.92 ± 0.32	59.51 ± 0.30
	CVAE	$\mathbf{61.63 \pm 0.65}$	$\mathbf{79.90 \pm 0.48}$	$\mathbf{41.97 \pm 0.32}$	$\mathbf{62.20 \pm 0.29}$
Vamp	MAML-CVAE	$\boxed{\mathbf{64.35 \pm 0.68}}$	$\mathbf{81.81 \pm 0.49}$	$\mathbf{44.46 \pm 0.32}$	$\mathbf{64.02 \pm 0.30}$
	CVAE	61.97 ± 0.68	80.64 ± 0.51	42.57 ± 0.32	62.79 ± 0.30
Pseudo	MAML-CVAE	$\boxed{\mathbf{64.98 \pm 0.66}}$	$\boxed{\mathbf{83.57 \pm 0.45}}$	$\boxed{\mathbf{45.46 \pm 0.33}}$	$\boxed{\mathbf{66.08 \pm 0.28}}$
	CVAE	62.44 ± 0.66	80.40 ± 0.49	42.47 ± 0.32	62.39 ± 0.30

Table 4: Accuracy in unsupervised few-shot classification as a downstream task of prototypical networks on Quickdraw between the MAML-CVAE and CVAE models. Accuracy is evaluated over N-way classification with S support and Q query images. Uncertainties are computed as 95% confidence interval over 1000 tasks. Bold font indicates best performing training algorithm. Boxes indicate overall best method or both if the confidence intervals overlap.

Comparison with other unsupervised approaches We compare our results to a common baseline for unsupervised few-shot classification that exploits meta-learning: the CACTUs model (Hsu et al., 2019). However, note that our model is not intended for unsupervised few-shot classification, but is rather used to evaluate the generated latent space. The CACTUs model automatically builds tasks from unlabelled data and then exploits meta-learning to find a learning procedure that allows to solve the constructed tasks. In order to construct tasks from unlabelled data it learns an appropriate embedding function mapping from the data space and then generates a partition over the embedded data-points leveraging k-means (MacQueen, 1967). The meta-learner is then trained over the so-defined labelled data through either the MAML algorithm ("CACTUs-MAML") or prototypical networks ("CACTUs-ProtoNets"). However, the learnt embedding may be used directly for downstream supervised learning. In particular, we consider the implementations with a linear classifier ("Emb. linear classifier") and a MLP with one hidden layer of 128 units ("Emb. MLP classifier"), which include the most performing ones. Two unsupervised embedding learning algorithms are used: ACAI (Berthelot et al., 2018) and BiGAN (Donahue et al., 2017; Dumoulin et al., 2017). In Table 5 we compare their results and their baselines against our most performing models in terms of few-shot classification accuracy.

$(N;\ S,Q)$	$(5;\ 1,19)$	$(5;\ 5,15)$	$(20;\ 1,19)$	$(20;\ 5,15)$
ACAI				
Emb. linear classifier	61.08 ± 1.32	81.82 ± 0.58	43.20 ± 0.69	66.33 ± 0.36
Emb. MLP classifier	51.95 ± 0.82	77.20 ± 0.65	30.65 ± 0.39	58.62 ± 0.41
CACTUs-ProtoNets	68.12 ± 0.84	83.58 ± 0.61	47.75 ± 0.43	66.27 ± 0.37
CACTUs-MAML	**68.84 ± 0.80**	**87.78 ± 0.50**	**48.09 ± 0.41**	**73.36 ± 0.34**
BiGAN				
Emb. linear classifier	48.28 ± 1.25	68.72 ± 0.66	27.80 ± 0.61	45.82 ± 0.37
Emb. MLP classifier	40.54 ± 0.79	62.56 ± 0.79	19.92 ± 0.32	40.71 ± 0.40
CACTUs-ProtoNets	54.74 ± 0.82	71.69 ± 0.73	33.40 ± 0.37	50.62 ± 0.39
CACTUs-MAML	**58.18 ± 0.81**	**78.66 ± 0.65**	**35.56 ± 0.36**	**58.62 ± 0.38**
ours				
CVAE - Pseudo	62.44 ± 0.66	80.40 ± 0.49	42.47 ± 0.32	62.39 ± 0.30
MAML-CVAE - Pseudo	**64.98 ± 0.66**	**83.57 ± 0.45**	**45.46 ± 0.33**	**66.08 ± 0.28**

Table 5: Accuracy comparison in few-shot classification tasks on Omniglot between our methods MAML-CVAE and CVAE and the CACTUs model together with its baselines (Hsu et al., 2019). The results are obtained as the average over 1000 tasks and uncertainties are computed as 95% confidence interval.

The comparison shows that the MAML-CVAE model implemented with Pseudo-inputs prior outperforms all BiGAN implementations. Furthermore, it achieves comparable accuracy to the ACAI implementation of CACTUs-ProtoNets but lower accuracy with respect to the ACAI implementation of CACTUs-MAML. We regard this as a significant result as our method was not meant for unsupervised few-shot classification.