VERSA: Versatile and Efficient Few-shot Learning

Jonathan Gordon[†*] **John Bronskill**[†*] **Matthias Bauer**[†,‡*]
Sebastian Nowozin[¶] **Richard E. Turner**[†¶]
[†]University of Cambridge [‡]Max Planck Institute for Intelligent Systems [¶]Microsoft Research
{jg801|jfb54|msb55|ret26}@cam.ac.uk Sebastian.Nowozin@microsoft.com

1 Introduction

Despite recent advances in few-shot learning, notably in meta-learning based approaches [Ravi and Larochelle, 2017, Vinyals et al., 2016, Edwards and Storkey, 2017, Finn et al., 2017, Lacoste et al., 2018], there remains a lack of general purpose methods for flexible, data-efficient learning. This paper introduces VERSA, a system for data efficient and versatile meta-learning. It employs a flexible and versatile amortization network that takes few-shot learning datasets as inputs, with arbitrary numbers of shots, and outputs a distribution over task-specific parameters in a single forward pass. VERSA substitutes optimization at test time with forward passes through inference networks, amortizing the cost of inference and relieving the need for second derivatives during training. We evaluate VERSA on benchmark datasets where the method achieves state-of-the-art results, handles arbitrary numbers of shots, and for classification, arbitrary numbers of classes at train and test time. The power of the approach is then demonstrated through a challenging few-shot ShapeNet view reconstruction task.

2 Meta-Learning Probabilistic Inference For Prediction

We now present the framework that consists of (i) a multi-task probabilistic model, and (ii) a method for meta-learning probabilistic inference.

2.1 Probabilistic Model

Two principles guide the choice of model. First, the use of discriminative models to maximize predictive performance on supervised learning tasks [Ng and Jordan, 2002]. Second, the need to leverage shared statistical structure between tasks (i.e. multi-task learning). These criteria are met by the standard multi-task directed graphical model shown in Fig. 1 that employs shared parameters θ, which are common to all tasks, and task specific parameters $\{\psi^{(t)}\}_{t=1}^{T}$. Inputs are denoted x and outputs y. Training data $D^{(t)} = \{(x_n^{(t)}, y_n^{(t)})\}_{n=1}^{N_t}$, and test data $\{(\tilde{x}_m^{(t)}, \tilde{y}_m^{(t)})\}_{m=1}^{M_t}$ are explicitly distinguished for each task t, as this is key for few-shot learning.

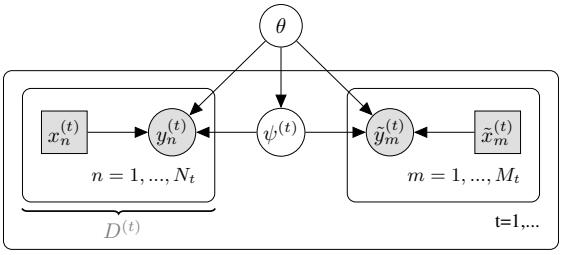


Figure 1: Directed graphical model for multi-task learning.

[*]Authors contributed equally.

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

Let $X^{(t)}$ and $Y^{(t)}$ denote all the inputs and outputs (both test and train) for task t. The joint probability of the outputs and task specific parameters for T tasks, given the inputs and global parameters is:

$$p\left(\{Y^{(t)}, \psi^{(t)}\}_{t=1}^{T} | \{X^{(t)}\}_{t=1}^{T}, \theta\right) = \prod_{t=1}^{T} p\left(\psi^{(t)}|\theta\right) \prod_{n=1}^{N_t} p\left(y_n^{(t)}|x_n^{(t)}, \psi^{(t)}, \theta\right) \prod_{m=1}^{M_t} p\left(\tilde{y}_m^{(t)}|\tilde{x}_m^{(t)}, \psi^{(t)}, \theta\right).$$

In the next section, the goal is to meta-learn fast and accurate approximations to the posterior predictive distribution $p(\tilde{y}^{(t)}|\tilde{x}^{(t)}, \theta) = \int p(\tilde{y}^{(t)}|\tilde{x}^{(t)}, \psi^{(t)}, \theta) p(\psi^{(t)}|\tilde{x}^{(t)}, D^{(t)}, \theta) \mathrm{d}\psi^{(t)}$ for unseen tasks t.

2.2 Probabilistic Inference

This section provides a framework for meta-learning approximate inference that is a simple reframing and extension of existing approaches [Finn et al., 2017, Grant et al., 2018]. We will employ point estimates for the shared parameters θ since data across all tasks will pin down their value. Distributional estimates will be used for the task-specific parameters since only a few shots constrain them.

Once the shared parameters are learned, the probabilistic solution to few-shot learning in the model above comprises two steps. First, form the posterior distribution over the task-specific parameters $p(\psi^{(t)}|\tilde{x}^{(t)}, D^{(t)}, \theta)$. Second, compute the posterior predictive $p(\tilde{y}^{(t)}|\tilde{x}^{(t)}, \theta)$. These steps will require approximation and the emphasis here is on performing this quickly at test time. We will describe the form of the approximation, the optimization problem used to learn it, and how to implement this efficiently below. In what follows we initially suppress dependencies on the inputs $\tilde{x}$ and shared parameters θ to reduce notational clutter, but will reintroduce these at the end of the section.

Specification of the approximate posterior predictive distribution. Our framework approximates the posterior predictive distribution by an amortized distribution $q_\phi(\tilde{y}|D)$. That is, we learn a feed-forward inference network with parameters ϕ that takes any training dataset $D^{(t)}$ and test input $\tilde{x}$ as inputs and returns the predictive distribution over the test output $\tilde{y}^{(t)}$. We construct this by amortizing the approximate posterior $q_\phi(\psi|D)$ and then form the approximate posterior predictive distribution using:

$$q_\phi(\tilde{y}|D) = \int p(\tilde{y}|\psi) q_\phi(\psi|D) \mathrm{d}\psi. \tag{1}$$

This step may require additional approximation e.g. Monte Carlo sampling. The amortization will enable fast predictions at test time. The form of these distributions is identical to those used in amortized variational inference [Edwards and Storkey, 2017, Kingma and Welling, 2014]. In this work, we use a factorized Gaussian distribution for $q_\phi(\psi|D^{(t)})$ with means and variances set by the amortization network. However, the training method described next is different.

Meta-learning the approximate posterior predictive distribution. The quality of the approximate posterior predictive for a single task will be measured by the KL-divergence between the true and approximate posterior predictive distribution $\mathrm{KL}\left[p(\tilde{y}|D) \| q_\phi(\tilde{y}|D)\right]$. The goal of learning will be to minimize the expected value of this KL averaged over tasks,

$$\phi^* = \operatorname*{argmin}_{\phi} \mathop{\mathbb{E}}_{p(D)} \left[\mathrm{KL}\left[p(\tilde{y}|D) \| q_\phi(\tilde{y}|D)\right]\right] = \operatorname*{argmax}_{\phi} \mathop{\mathbb{E}}_{p(\tilde{y},D)} \left[\log \int p(\tilde{y}|\psi) q_\phi(\psi|D) \mathrm{d}\psi\right]. \tag{2}$$

Training will therefore return parameters ϕ that best approximate the posterior predictive distribution in an average KL sense. So, if the approximate posterior $q_\phi(\psi|D)$ is rich enough, *global* optimization will recover the true posterior $p(\psi|D)$ (assuming $p(\psi|D)$ obeys identifiability conditions [Casella and Berger, 2002]).[2] Thus, the amortized procedure meta-learns approximate inference that supports accurate prediction.

The right hand side of Eq. (2) indicates how training could proceed: (i) select a task t at random, (ii) sample some training data $D^{(t)}$, (iii) form the posterior predictive $q_\phi(\cdot|D^{(t)})$ and, (iv) compute the log-density $\log q_\phi(\tilde{y}^{(t)}|D^{(t)})$ at test data $\tilde{y}^{(t)}$ *not included in* $D^{(t)}$. Repeating this process many times and averaging the results would provide an unbiased estimate of the objective which can then be optimized. This perspective also makes it clear that the procedure is scoring the approximate

[2]Note that the true *predictive* posterior $p(y|D)$ is recovered regardless of the identifiability of $p(\psi|D)$.

inference procedure by simulating approximate Bayesian held-out log-likelihood evaluation. Importantly, while an inference network is used to approximate posterior distributions, the training procedure differs significantly from standard variational inference. In particular, rather than minimizing $\mathrm{KL}(q_\phi(\psi|D)\|p(\psi|D))$, our objective function directly focuses on the posterior predictive distribution and minimizes $\mathrm{KL}(p(\tilde{y}|D)\|q_\phi(\tilde{y}|D))$.

End-to-end stochastic training. Armed by the insights above we now layout the full training procedure. We reintroduce inputs and shared parameters θ and the objective becomes:

$$\mathcal{L}(\phi) = -\mathop{\mathbb{E}}_{p(D,\tilde{y},\tilde{x})}\left[\log q_\phi(\tilde{y}|\tilde{x},\theta)\right] = -\mathop{\mathbb{E}}_{p(D,\tilde{y},\tilde{x})}\left[\log \int p(\tilde{y}|\tilde{x},\psi,\theta) q_\phi(\psi|D,\theta)\mathrm{d}\psi\right]. \tag{3}$$

We optimize the objective over the shared parameters θ as this will maximize predictive performance (i.e., Bayesian held out likelihood). An end-to-end stochastic training objective for θ and ϕ is:

$$\hat{\mathcal{L}}(\theta,\phi) = \frac{1}{MT}\sum_{M,T}\log\frac{1}{L}\sum_{l=1}^{L} p\left(\tilde{y}_m^{(t)}|\tilde{x}_m^{(t)},\psi_l^{(t)},\theta\right), \quad \text{with } \psi_l^{(t)} \sim q_\phi(\psi|D^{(t)},\theta) \tag{4}$$

and $\{\tilde{y}_m^{(t)},\tilde{x}_m^{(t)},D^{(t)}\} \sim p(\tilde{y},\tilde{x},D)$, where p represents the data distribution (e.g., sampling tasks and splitting them into disjoint training data D and test data $\{(\tilde{x}_m^{(t)},\tilde{y}_m^{(t)})\}_{m=1}^{M_t}$). This type of training therefore uses episodic train / test splits at meta-train time. We have also approximated the integral over ψ using L Monte Carlo samples. The local reparametrization [Kingma et al., 2015] trick enables optimization. Interestingly, the learning objective does not require an explicit specification of the prior distribution over parameters, $p(\psi^{(t)}|\theta)$, learning it implicitly through $q_\phi(\psi|D,\theta)$ instead.

3 Versatile Amortized Inference

A versatile system is one that makes inferences both rapidly *and* flexibly. By rapidly we mean that test-time inference involves only simple computation such as a feed-forward pass through a neural network. By flexibly we mean that the system supports a variety of tasks – including variable numbers of shots or numbers of classes in classification problems – without retraining. Rapid inference comes automatically with the use of a deep neural network to amortize the approximate posterior distribution q. However, it typically comes at the cost of flexibility: amortized inference is usually limited to a single specific task. Below, we discuss design choices that enable us to retain flexibility.

Inference with sets as inputs. The amortization network takes data sets of variable size as inputs whose ordering we should be invariant to. We use permutation-invariant *instance-pooling* operations to process these sets similarly to Qi et al. [2017] and as formalized in Zaheer et al. [2017]. The instance-pooling operation ensures that the network can process any number of training observations.

VERSA for Few-Shot Image Classification. For few-shot image classification, our parameterization of the probabilistic model is inspired by early work from Heskes [2000], Bakker and Heskes [2003] and recent extensions to deep learning [Bauer et al., 2017, Qiao et al., 2017]. A feature extractor neural network $h_\theta(x) \in \mathbb{R}^{d_\theta}$, shared across all tasks, feeds into a set of task-specific linear classifiers with softmax outputs and weights and biases $\psi^{(t)} = \{W^{(t)}, b^{(t)}\}$ (see Fig. 2).

A naive amortization requires the approximate posterior $q_\phi(\psi|D,\theta)$ to model the distribution over full weight matrices in $\mathbb{R}^{d_\theta \times C}$ (and biases). This requires the specification of the number of few-shot classes C ahead of time and limits inference to this chosen number. Moreover, it is difficult to meta-learn systems that directly output large matrices as the output dimensionality is high. We therefore propose specifying $q_\phi(\psi|D,\theta)$ in a *context independent* manner such that each weight vector ψ_c depends only on examples from class c, by amortizing individual weight vectors associated with a single softmax output instead of the entire weight matrix directly. To reduce the number of learned parameters, the amortization network operates directly on the extracted features $h_\theta(x)$:

$$q_\phi(\psi|D,\theta) = \prod_{c=1}^{C} q_\phi\left(\psi_c|\{h_\theta(x_n^c)\}_{n=1}^{k_c},\theta\right). \tag{5}$$

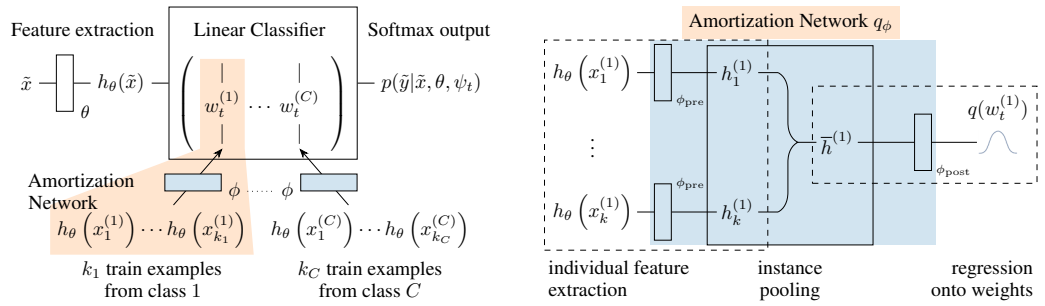


Figure 2: Computational flow of VERSA for few-shot classification with the context-independent approximation. *Left:* A test point $\tilde{x}$ is mapped to its softmax output through a feature extractor neural network and a linear classifier (fully connected layer). The global parameters θ of the feature extractor are shared between tasks whereas the weight vectors $w_t^{(c)}$ of the linear classifier are task specific and inferred through an amortization network with parameters ϕ. *Right:* Amortization network that maps the extracted features of the k training examples of a particular class to the corresponding weight vector of the linear classifier.

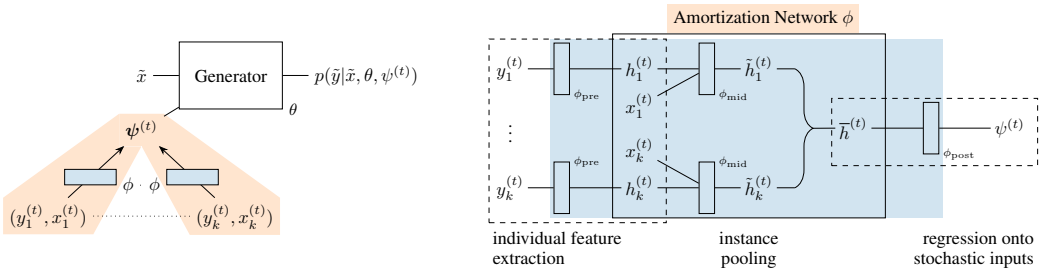


Figure 3: Computational flow of VERSA for few-shot view reconstruction. *Left:* A set of training images and angles $\{(y_n^{(t)}, x_n^{(t)})\}_{n=1}^k$ are mapped to a stochastic input $\psi^{(t)}$ through the amortization network q_ϕ. $\psi^{(t)}$ is then concatenated with a test angle $\tilde{x}$ and mapped onto a new image through the generator θ. *Right:* Amortization network that maps k image/angle examples of a particular object-instance to the corresponding stochastic input.

Note that in our implementation, end-to-end training is employed, i.e., we backpropagate to θ through the inference network. Here k_c is the number of observed examples in class c and $\psi_c = \{w_c, b_c\}$ denotes the weight vector and bias of the linear classifier associated with that class. Thus, we construct the classification matrix $\psi^{(t)}$ by performing C feed-forward passes through the inference network $q_\phi(\psi|D, \theta)$ (see Fig. 2).

The assumption of context independent inference is an approximation. Our theoretical arguments use insights from Density Ratio Estimation [Mohamed, 2018, Sugiyama et al., 2012], and we empirically demonstrate that full approximate posterior distributions are close to their context independent counterparts. Critically, the context independent approximation addresses all the limitations of a naive amortization mentioned above: (i) the inference network needs to amortize far fewer parameters whose number does not scale with number of classes C (a single weight vector instead of the entire matrix); (ii) the amortization network can be meta-trained with different numbers of classes per task, and (iii) the number of classes C can vary at test-time.

VERSA for Few-Shot Image Reconstruction (Regression). We consider a challenging few-shot learning task with a complex (high dimensional and continuous) output space. We define view reconstruction as the ability to infer how an object looks from any desired angle based on a small set of observed views. We frame this as a multi-output regression task from a set of training images with known orientations to output images with specified orientations.

Our generative model is similar to the generator of a GAN or the decoder of a VAE: A latent vector $\psi^{(t)} \in \mathbb{R}^{d_\psi}$, which acts as an object-instance level input to the generator, is concatenated with an angle representation and mapped through the generator to produce an image at the specified orientation. In this setting, we treat all parameters θ of the generator network as global parameters (see Appendix E for full details of the architecture), whereas the latent inputs $\psi^{(t)}$ are the task-specific parameters. We

use a Gaussian likelihood in pixel space for the outputs of the generator. To ensure that the output means are between zero and one, we use a sigmoid activation after the final layer. ϕ parameterizes an amortization network that first processes the image representations of an object, concatenates them with their associated view orientations, and processes them further before instance-pooling. From the pooled representations, $q_\phi(\psi|D,\theta)$ produces a distribution over vectors $\psi^{(t)}$. This process is illustrated in Fig. 3. Note that this approach to view construction is conceptually similar to Generative Query Networks [Eslami et al., 2018].

4 Experiments and Results

4.1 Few Shot Classification

We evaluate VERSA on standard few-shot classification tasks in comparison to previous work. Specifically, we consider the Omniglot [Lake et al., 2011] and *mini*ImageNet [Ravi and Larochelle, 2017] datasets which are C-way classification tasks with k_c examples per class. VERSA follows the implementation in Sections 2 and 3, and the approximate inference scheme in Eq. (5). We follow the experimental protocol established by Vinyals et al. [2016] for Omniglot and Ravi and Larochelle [2017] for *mini*Imagenet, using equivalent architectures for h_θ. Training is carried out in an episodic manner: for each task, k_c examples are used as training inputs to infer $q_\phi(\psi^{(c)}|D,\theta)$ for each class, and an additional set of examples is used to evaluate the objective function.

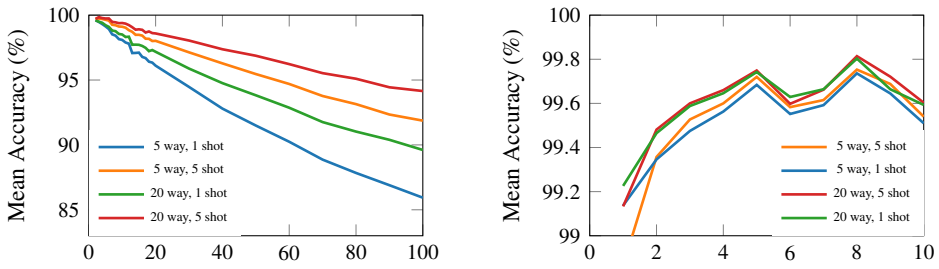
Table 1 details few-shot classification performance for VERSA as well as competitive approaches. The tables include results for only those approaches with comparable training procedures and convolutional feature extraction architectures. Approaches that employ pre-training and/or residual networks [Bauer et al., 2017, Qiao et al., 2017, Rusu et al., 2018, Gidaris and Komodakis, 2018, Oreshkin et al., 2018, Garcia and Bruna, 2017, Lacoste et al., 2018] have been excluded so that the quality of the learning algorithm can be assessed separately from the power of the underlying discriminative model.

For Omniglot, the training, validation, and test splits have not been specified for previous methods, affecting the comparison. VERSA achieves a new state-of-the-art results (67.37% - up 1.38% over the previous best) on 5-way - 5-shot classification on the *mini*ImageNet benchmark and (97.66% - up 0.02%) on the 20-way - 1 shot Omniglot benchmark for systems using a convolution-based network architecture and an end-to-end training procedure. VERSA is within error bars of state-of-the-art on three other benchmarks including 5-way - 1-shot *mini*ImageNet, 5-way - 5-shot Omniglot, and 5-way - 1-shot Omniglot. Results on the Omniglot 20 way - 5-shot benchmark are very competitive with, but lower than other approaches. While most of the methods evaluated in Table 1 adapt all of the learned parameters for new tasks, VERSA is able to achieve state-of-the-art performance despite adapting only the weights of the top-level classifier.

Table 1: Accuracy results for few-shot classification. The $\pm$ sign indicates the 95% confidence interval. Bold text indicates the highest scores that overlap in their confidence intervals.

	Omniglot				*miniImageNet*	
	5-way accuracy (%)		**20-way accuracy (%)**		**5-way accuracy (%)**	
Method	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Siamese Nets [Koch et al., 2015]	97.3	98.4	88.1	97.0		
Matching Nets [Vinyals et al., 2016]	98.1	98.9	93.8	98.5	46.6	60.0
Neural Statistician [Edwards and Storkey, 2017]	98.1	99.5	93.2	98.1		
Memory Mod [Kaiser et al., 2017]	98.4	99.6	95.0	98.6		
Meta LSTM [Ravi and Larochelle, 2017]					43.44 ± 0.77	60.60 ± 0.71
MAML [Finn et al., 2017]	98.7 ± 0.4	$\mathbf{99.9 \pm 0.1}$	95.8 ± 0.3	98.9 ± 0.2	48.7 ± 1.84	63.11 ± 0.92
Prototypical Nets [Snell et al., 2017]	97.4	99.3	95.4	98.7	46.61 ± 0.78	65.77 ± 0.70
mAP-SSVM [Triantafillou et al., 2017]	98.6	99.6	95.2	98.6	50.32 ± 0.80	63.94 ± 0.72
mAP-DLM [Triantafillou et al., 2017]	98.8	99.6	95.4	98.6	50.28 ± 0.80	63.70 ± 0.70
LLAMA [Grant et al., 2018]					49.40 ± 1.83	
PLATIPUS [Finn et al., 2018]					50.13 ± 1.86	
Meta-SGD [Li et al., 2017]	$\mathbf{99.53 \pm 0.26}$	$\mathbf{99.93 \pm 0.09}$	95.93 ± 0.38	98.97 ± 0.19	50.47 ± 1.87	64.03 ± 0.94
SNAIL [Mishra et al., 2018]	99.07 ± 0.16	99.78 ± 0.09	$\mathbf{97.64 \pm 0.30}$	$\mathbf{99.36 \pm 0.18}$	45.1	55.2
Relation Net [Yang et al., 2018]	$\mathbf{99.6 \pm 0.2}$	$\mathbf{99.8 \pm 0.1}$	$\mathbf{97.6 \pm 0.2}$	$\mathbf{99.1 \pm 0.1}$	50.44 ± 0.82	65.32 ± 0.70
Reptile [Nichol and Schulman, 2018]	97.68 ± 0.04	99.48 ± 0.06	89.43 ± 0.14	97.12 ± 0.32	49.97 ± 0.32	$\mathbf{65.99 \pm 0.58}$
BMAML [Kim et al., 2018]					$\mathbf{53.8 \pm 1.46}$	
VERSA (Ours)	$\mathbf{99.70 \pm 0.20}$	$\mathbf{99.75 \pm 0.13}$	$\mathbf{97.66 \pm 0.29}$	98.77 ± 0.18	$\mathbf{53.40 \pm 1.82}$	$\mathbf{67.37 \pm 0.86}$

Versatility. VERSA allows us to vary the number of classes C and shots k_c between training and testing (Eq. (5)). Fig. 4a shows that a model trained for a particular C-way retains very high accuracy as C is varied. For example, when VERSA is trained for the 20-Way, 5-Shot condition, at test-time it can handle $C = 100$ way conditions and retain an accuracy of approximately 94%. Fig. 4b shows similar robustness as the number of shots k_c is varied. VERSA therefore demonstrates considerable flexibility and robustness to the test-time conditions, but at the same time it is efficient as it only requires forward passes through the network. The time taken to evaluate 1000 test tasks with a 5-way, 5-shot *mini*ImageNet trained model using MAML (`https://github.com/cbfinn/maml`) is 302.9 seconds whereas VERSA took 53.5 seconds on a NVIDIA Tesla P100-PCIE-16GB GPU. This is more than $5\times$ speed advantage in favor of VERSA while bettering MAML in accuracy by 4.26%.



(a) Way (C) **(b)** Shot (k_c)

Figure 4: Test accuracy on Omniglot when varying (a) way (fixing shot to be that used for training) and (b) shot. In Fig. 4b, all models are evaluated on 5-way classification. Colors indicate models trained with different way-shot episodic combinations.

4.2 Few-Shot View Reconstruction

ShapeNetCore v2 [Chang et al., 2015] is a database of 3D objects covering 55 common object categories with ∼51,300 unique objects. For our experiments, we use 12 of the largest object categories. We concatenate all instances from all 12 of the object categories together to obtain a dataset of 37,108 objects. This dataset is then randomly shuffled and we use 70% of the objects for training, 10% for validation, and 20% for testing. For each object, we generate 36 views of size 32×32 pixels spaced evenly every 10 degrees in azimuth around the object.

We evaluate VERSA by comparing it to a conditional variational autoencoder (C-VAE) with view angles as labels [Kingma et al., 2014, Narayanaswamy et al., 2017] and identical architectures. We train VERSA in an episodic manner and the C-VAE in batch-mode on all 12 object classes at once. We train on a single view selected at random and use the remaining views to evaluate the objective function. Fig. 5 shows views of unseen objects from the test set generated from a single shot with VERSA as well as a C-VAE and compares both to ground truth views. Both VERSA and the C-VAE capture the correct orientation of the object in the generated images. However, VERSA produces images that contain much more detail and are visually sharper than the C-VAE images. Although important information is missing due to occlusion in the single shot, VERSA is often able to accurately impute this information presumably due to learning the statistics of these objects.

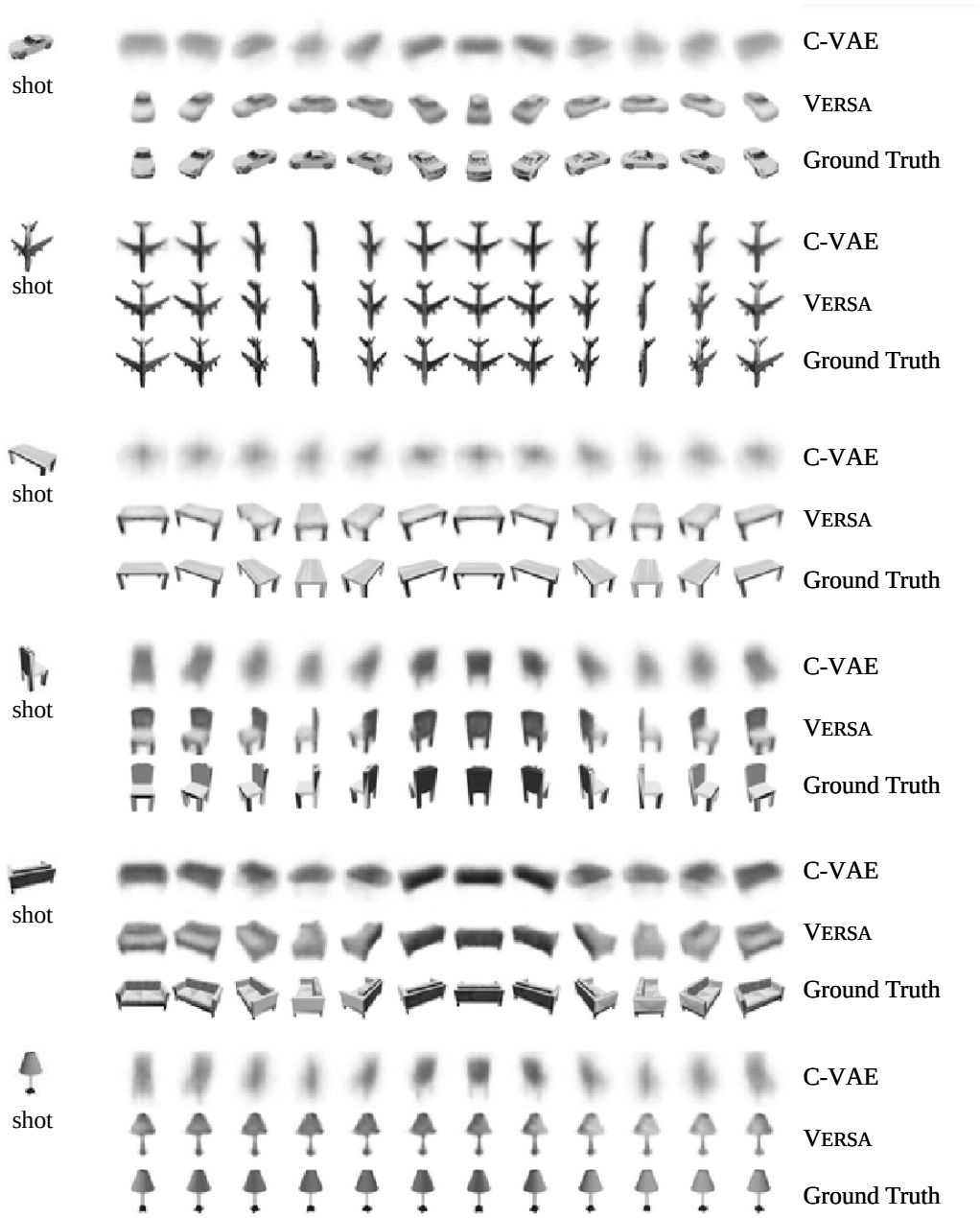


Figure 5: Results for ShapeNet view reconstruction for unseen objects from the test set (shown left). The model was trained to reconstruct views from a single orientation. *Top row:* images/views generated by a C-VAE model; *middle row* images/views generated by VERSA; *bottom row:* ground truth images. Views are spaced evenly every 30 degrees in azimuth.

References

B. Bakker and T. Heskes. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research*, 4(May):83–99, 2003.

M. Bauer, M. Rojas-Carulla, J. B. Świątkowski, B. Schölkopf, and R. E. Turner. Discriminative k-shot learning using probabilistic models. *arXiv preprint arXiv:1706.00326*, 2017.

G. Casella and R. L. Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

H. Edwards and A. Storkey. Towards a neural statistician. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

S. M. A. Eslami, D. Jimenez Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, D. P. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. Rabinowitz, H. King, C. Hillier, M. Botvinick, D. Wierstra, K. Kavukcuoglu, and D. Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. ISSN 0036-8075. doi: 10.1126/science.aar6170. URL `http://science.sciencemag.org/content/360/6394/1204`.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017.

C. Finn, K. Xu, and S. Levine. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

V. Garcia and J. Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018.

E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths. Recasting gradient-based meta-learning as hierarchical Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

T. Heskes. Empirical bayes for learning to learn. 2000.

Ł. Kaiser, O. Nachum, R. Aurko, and S. Bengio. Learning to remember rare events. In *International Conference on Learning Representations (ICLR)*, 2017.

T. Kim, J. Yoon, O. Dia, S. Kim, Y. Bengio, and S. Ahn. Bayesian model-agnostic meta-learning. *arXiv preprint arXiv:1806.03836*, 2018.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

A. Lacoste, B. Oreshkin, W. Chung, T. Boquet, N. Rostamzadeh, and D. Krueger. Uncertainty in multitask transfer learning. *arXiv preprint arXiv:1806.07528*, 2018.

B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33, 2011.

Z. Li, F. Zhou, F. Chen, and H. Li. Meta-sgd: Learning to learn quickly for few shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel. A simple neural attentive meta-learner. 2018.

S. Mohamed. Density ratio trick. `http://blog.shakirm.com/2018/01/machine-learning-trick-of-the-day-7-density-ratio-trick/`, 2018.

S. Narayanaswamy, T. B. Paige, J.-W. van de Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, and P. Torr. Learning disentangled representations with semi-supervised deep generative models. In *Advances in Neural Information Processing Systems*, pages 5927–5937, 2017.

A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems*, pages 841–848, 2002.

A. Nichol and J. Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2018.

B. N. Oreshkin, A. Lacoste, and P. Rodriguez. Tadam: Task dependent adaptive metric for improved few-shot learning. *arXiv preprint arXiv:1805.10123*, 2018.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

S. Qiao, C. Liu, W. Shen, and A. Yuille. Few-shot image recognition by predicting parameters from activations. *arXiv preprint arXiv:1706.03466*, 2017.

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

A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4080–4090, 2017.

M. Sugiyama, T. Suzuki, and T. Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

E. Triantafillou, R. Zemel, and R. Urtasun. Few-shot learning through an information retrieval lens. In *Advances in Neural Information Processing Systems*, pages 2255–2265, 2017.

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

F. S. Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. 2018.

M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, pages 3394–3404, 2017.