Supplementary Materials for "Variational Multi-Task Learning with Gumbel-Softmax Priors"

Jiayi Shen[1], **Xiantong Zhen**[1,2], **Marcel Worring**[1], **Ling Shao**[2]
[1]AIM Lab, University of Amsterdam, Netherlands
[2]Inception Institute of Artificial Intelligence, Abu Dhabi, UAE

A Derivation

A.1 Evidence Lower Bound for Multi-Task Learning

In this paper, we follow the multi-input multi-output data setting [33, 52] for multi-task learning, each task t has its own training data $\mathcal{D}_t = \{\mathbf{x}_{t,n}, \mathbf{y}_{t,n}\}_{n=1}^{N_t}$. Note that we derive our methodology mainly using terminologies related to classification tasks, but it is also applicable to regression tasks.

Under the probabilistic formulation for multi-task learning, we start with the conditional log-likelihood for each task t: $\log p(\mathbf{y}_{t,n}|\mathbf{x}_{t,n}, \mathcal{D}_{1:T\setminus t})$, where $(\mathbf{x}_{t,n}, \mathbf{y}_{t,n})$ is a sample from the data of current task t and $\mathcal{D}_{1:T\setminus t}$ is the data from all related tasks.

Here, we introduce the latent variables $\mathbf{z}_{t,n}$ and $\mathbf{w}_t$:

$$\log p(\mathbf{y}_{t,n}|\mathbf{x}_{t,n}, \mathcal{D}_{1:T\setminus t}) = \log \int\int p(\mathbf{y}_{t,n}, \mathbf{z}_{t,n}, \mathbf{w}_t, |\mathbf{x}_{t,n}, \mathcal{D}_{1:T\setminus t}) d\mathbf{z}_{t,n} d\mathbf{w}_t, \tag{1}$$

where $p(\mathbf{y}_{t,n}, \mathbf{z}_{t,n}, \mathbf{w}_t, |\mathbf{x}_{t,n}, \mathcal{D}_{1:T\setminus t})$ is the joint conditional predictive distribution over the classification label or regression target. Under the assumption that $\mathbf{w}_t$ and $\mathbf{z}_{t,n}$ are conditionally independent, we obtain

$$\log p(\mathbf{y}_{t,n}|\mathbf{x}_{t,n}, \mathcal{D}_{1:T\setminus t}) = \log \int\int p(\mathbf{y}_{t,n}|\mathbf{z}_{t,n}, \mathbf{w}_t) p(\mathbf{z}_{t,n}|\mathbf{x}_{t,n}, \mathcal{D}_{1:T\setminus t}) p(\mathbf{w}_t|\mathcal{D}_{1:T\setminus t}) d\mathbf{z}_{t,n} d\mathbf{w}_t. \tag{2}$$

Next, we introduce the variational posteriors $q_\phi(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})$ and $q_\theta(\mathbf{w}_t|\mathcal{D}_t)$ to approximate the true posteriors for latent representations and classifiers, respectively. By leveraging Jensen's inequality, we have the following steps as

$$\begin{aligned}
&\log p(\mathbf{y}_{t,n}|\mathbf{x}_{t,n}, \mathcal{D}_{1:T\setminus t}) \\
&= \log \int\int p(\mathbf{y}_{t,n}|\mathbf{z}_{t,n}, \mathbf{w}_t) p(\mathbf{z}_{t,n}|\mathbf{x}_{t,n}, \mathcal{D}_{1:T\setminus t}) d\mathbf{z}_{t,n} p(\mathbf{w}_t|\mathcal{D}_{1:T\setminus t}) \frac{q_\theta(\mathbf{w}_t|\mathcal{D}_t)}{q_\theta(\mathbf{w}_t|\mathcal{D}_t)} d\mathbf{w}_t \\
&\geq \int \log \Big[\frac{\int p(\mathbf{y}_{t,n}|\mathbf{z}_{t,n}, \mathbf{w}_t) p(\mathbf{z}_{t,n}|\mathbf{x}_{t,n}, \mathcal{D}_{1:T\setminus t}) d\mathbf{z}_{t,n} p(\mathbf{w}_t|\mathcal{D}_{1:T\setminus t})}{q_\theta(\mathbf{w}_t|\mathcal{D}_t)} \Big] q_\theta(\mathbf{w}_t|\mathcal{D}_t) d\mathbf{w}_t \\
&= \mathbb{E}_{q_\theta} \big[\log \int p(\mathbf{y}_{t,n}|\mathbf{z}_{t,n}, \mathbf{w}_t) p(\mathbf{z}_{t,n}|\mathbf{x}_{t,n}, \mathcal{D}_{1:T\setminus t}) \frac{q_\phi(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})}{q_\phi(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})} d\mathbf{z}_{t,n} \big] \\
&\quad - \mathbb{KL}[q_\theta(\mathbf{w}_t|\mathcal{D}_t)||p(\mathbf{w}_t|\mathcal{D}_{1:T\setminus t})] \\
&\geq \mathbb{E}_{q_\theta} \mathbb{E}_{q_\phi} [\log p(\mathbf{y}_{t,n}|\mathbf{z}_{t,n}, \mathbf{w}_t)] - \mathbb{KL}[q_\phi(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})||p(\mathbf{z}_{t,n}|\mathbf{x}_{t,n}, \mathcal{D}_{1:T\setminus t})] \\
&\quad - \mathbb{KL}[q_\theta(\mathbf{w}_t|\mathcal{D}_t)||p(\mathbf{w}_t|\mathcal{D}_{1:T\setminus t})].
\end{aligned} \tag{3}$$

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Thus, we obtain the ELBO for multi-task learning with latent representations and classifiers as follows:

$$\frac{1}{T}\sum_{t=1}^{T}\log p(\mathcal{Y}_t|\mathcal{X}_t,\mathcal{D}_{1:T\setminus t}) \geq \frac{1}{T}\sum_{t=1}^{T}\Bigg\{\sum_{n=1}^{N_t}\Big\{\mathbb{E}_{q_\theta}\mathbb{E}_{q_\phi}[\log p(\mathbf{y}_{t,n}|\mathbf{z}_{t,n},\mathbf{w}_t)] \\ -\mathbb{KL}[q_\phi(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})||p(\mathbf{z}_{t,n}|\mathbf{x}_{t,n},\mathcal{D}_{1:T\setminus t})]\Big\} - \mathbb{KL}[q_\theta(\mathbf{w}_t|\mathcal{D}_t)||p(\mathbf{w}_t|\mathcal{D}_{1:T\setminus t})]\Bigg\}. \tag{4}$$

We integrate this ELBO with the proposed Gumbel-Softmax priors to obtain the empirical objective for variational multi-task learning. To verify the effectiveness of our proposed models, we define the basic variational Bayesian multi-task learning (VBMTL) as a baseline. VBMTL shares the inference network of latent representations among tasks but just applies the normal Gaussian as priors of the latent variables.

In this paper, we propose variational multi-task learning, a general probabilistic inference framework, in which we cast multi-task learning as a variational Bayesian inference problem. This general framework can be seamlessly combined with the advantages of other deterministic approaches in leveraging shared knowledge among tasks. We can in fact take advantage of deterministic approaches to generalize even better in more settings, e.g., large amounts of training data. In this case, we can train a large convolutional neural network fully end-to-end to extract more representative features specifically for individual tasks.

A.2 Evidence Lower Bound for Single-Task Learning

Generally, the proposed Bayesian inference framework which infers the posteriors of presentations $\mathbf{z}$ and classifiers $\mathbf{w}$ jointly can be widely applied in other research fields. For example, based on the proposed Bayesian inference framework, we introduce a variational version of single-task learning (VSTL) and provide the derivation of its evidence lower bound. It is worth noting that single-task learning does not share knowledge among tasks, thus both inference networks of latent representations and classifiers are task-specific. And the log-likelihood for single-task learning is not allowed to be conditioned on the data from other tasks.

$$\begin{aligned}
&\log p(\mathbf{y}_{t,n}|\mathbf{x}_{t,n}) \\
&= \log\int\int p(\mathbf{y}_{t,n}|\mathbf{z}_{t,n},\mathbf{w}_t)p(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})p(\mathbf{w}_t)d\mathbf{z}_{t,n}d\mathbf{w}_t \\
&= \log\int\int p(\mathbf{y}_{t,n}|\mathbf{z}_{t,n},\mathbf{w}_t)p(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})d\mathbf{z}_{t,n}p(\mathbf{w}_t)\frac{q_\theta(\mathbf{w}_t)}{q_\theta(\mathbf{w}_t)}d\mathbf{w}_t \\
&\geq \int\log[\frac{\int p(\mathbf{y}_{t,n}|\mathbf{z}_{t,n},\mathbf{w}_t)p(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})d\mathbf{z}_{t,n}p(\mathbf{w}_t)}{q_\theta(\mathbf{w}_t)}]q_\theta(\mathbf{w}_t)d\mathbf{w}_t \\
&= \mathbb{E}_{q_\theta}\big[\log\int p(\mathbf{y}_{t,n}|\mathbf{z}_{t,n},\mathbf{w}_t)p(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})\frac{q_\phi(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})}{q_\phi(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})}d\mathbf{z}_{t,n}\big] - \mathbb{KL}[q_\theta(\mathbf{w}_t)||p(\mathbf{w}_t)] \\
&\geq \mathbb{E}_{q_\theta}\mathbb{E}_{q_\phi}[\log p(\mathbf{y}_{t,n}|\mathbf{z}_{t,n},\mathbf{w}_t)] - \mathbb{KL}[q_\phi(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})||p(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})] - \mathbb{KL}[q_\theta(\mathbf{w}_t)||p(\mathbf{w}_t)].
\end{aligned} \tag{5}$$

In this case, tasks are learned independently with no access to shared knowledge provided by other tasks, thus the priors $p(\mathbf{w}_t)$ and $p(\mathbf{z}_{t,n}|\mathbf{x}_{t,n})$ are set to normal Gaussians as applied in [**? ? ?**].

B More Experimental Details

We train all models and parameters by the Adam optimizer [27] using an NVIDIA Tesla V100 GPU. The learning rate is initially set as $1e-4$ and decreases with a factor of 0.5 every $3K$ iterations. Details of iteration numbers and batch sizes for different benchmarks are provided in Table B.1. In each batch, the number of training samples from each task and category is identical. The network architectures of our methods for the four benchmarks are given. The code will be available at https://github.com/autumn9999/VMTL.git.

Table B.1. The iteration numbers and batch sizes on different datasets, where C and T denotes the number of classes and tasks in the specific dataset, respectively.

Dataset	Iteration	Batch size
Office-Home	$15,000$	$4 * C * T$
Office-Caltech	$15,000$	$4 * C * T$
ImageCLEF	$15,000$	$4 * C * T$
DomainNet	$30,000$	$2 * C * T$

Table B.2. The inference network $\theta(\cdot)$ for amortized classifiers in VMTL-AC.

Output size	Layers
4096	Input feature
4096	Dropout (p=0.7)
512	Fully connected, ELU
512	Fully connected, ELU
512	Local reparameterization to μ_w, σ_w^2

Table B.3. The inference network $\phi(\cdot)$ for latent representations.

Output size	Layers
$4096, N * 4096$	Input features
4096	Cross attention
4096	Dropout (p=0.7)
512	Fully connected, ELU
512	Fully connected, ELU
512	Local reparameterization to μ_z, σ_z^2

B.1 Inference Networks

The architecture of the inference network for amortized classifiers in VMTL-AC is in Table B.2. In VMTL, we directly learn the parameters of the distribution of variational posteriors, which has the same dimension as the latent representation. The architecture of the inference network for latent representations is in Table B.3. During inference, we apply the reparameterization trick to generate the samples for both latent variables [**?**].

B.2 Implementation of the Compared Previous Works

In this paper, we compare against four representative methods [3, 25, 33, 38], which are implemented by following the same experimental setup as our methods. In practice, we implement the method, Long et al. [33], by applying its open code repository (https://github.com/thuml/MTlearn) under the same experimental environments as ours. For other compared methods, the models consist a shared feature extractor and task-specific classifiers. In particular, Bakker et al. [3] is a Bayesian method for multi-task learning optimized by an expectation-maximization algorithm, producing very competitive performance. Kendall et al. [25] propose to weigh multiple loss functions by considering the homoscedastic uncertainty of each task, which shows the benefit of modeling uncertainty. Qian et al. [38] integrate the variational information bottleneck [**?**] to the method based on [25], which shows the benefit of exploring shared information for latent representations.

C More Experimental Results

C.1 Effectiveness in Handling Limited Data

The results of average accuracy on the *Office-Home*, *Office-Caltech*, and *ImageCLEF* datasets are given in Tables C.4, C.5, and C.6, respectively, which provide detailed information for Fig. 2 of the paper. Our proposed probabilistic models, i.e., VMTL and VMTL-AC outperform the deterministic baseline multi-task learning model (BMTL), which demonstrates the benefits of our proposed

Table C.4. Performance under different proportions of training data on *Office-Home*.

Methods	5%	10%	20%	40%	60%
STL	49.2±0.2	58.3±0.1	64.9±0.1	70.3±0.2	73.4±0.1
VSTL	51.1±0.1	60.2±0.2	65.8±0.2	**72.4±0.3**	73.8±0.2
BMTL	50.4±0.1	59.5±0.1	65.6±0.1	70.5±0.1	69.5±0.2
VBMTL	51.3±0.1	60.9±0.1	67.0±0.2	72.1±0.1	74.0±0.1
VMTL-AC	56.3±0.1	63.8±0.1	68.3±0.1	69.0±0.1	73.7±0.2
VMTL	**58.3±0.1**	**65.0±0.0**	**69.2±0.2**	71.5±0.3	**74.2±0.1**

Table C.5. Performance under different proportions of training data on *Office-Caltech*.

Methods	5%	10%	20%	40%	60%
STL	88.6±0.3	90.7±0.2	92.4±0.3	96.6±0.2	97.2±0.2
VSTL	89.0±0.2	91.1±0.2	93.4±0.3	96.7±0.2	97.1±0.3
BMTL	89.5±0.3	92.3±0.2	93.1±0.1	95.4±0.3	97.0±0.2
VBMTL	90.8±0.6	93.2±0.2	93.5±0.1	96.5±0.1	97.1±0.4
VMTL-AC	**93.9±0.1**	95.1±0.0	**95.2±0.1**	**96.8±0.2**	97.2±0.2
VMTL	93.8±0.1	**95.3±0.0**	**95.2±0.1**	96.5±0.2	**97.3±0.1**

Table C.6. Performance under different proportions of training data on *ImageCLEF*.

Methods	5%	10%	20%	40%	60%
STL	62.6±0.2	69.7±0.3	76.2±0.3	79.3±0.2	80.1±0.1
VSTL	64.9±0.3	70.8±0.3	77.2±0.2	80.3±0.1	80.8±0.1
BMTL	65.7±0.4	72.0±0.3	76.8±0.3	79.0±0.3	80.5±0.2
VBMTL	67.1±0.3	73.0±0.7	78.0±0.2	81.2±0.1	81.0±0.2
VMTL-AC	75.7±0.3	77.6±0.2	80.0±0.1	81.3±0.2	**82.3±0.3**
VMTL	**76.2±0.3**	**77.9±0.2**	**80.2±0.1**	**82.4±0.4**	**82.3±0.2**

variational Bayesian framework. Given a limited amount of training data, our models also have a better performance than VBMTL, which demonstrates that the proposed Gumbel-Softmax priors are beneficial to fully leverage the shared knowledge among tasks. When training data is limited, STL and VSTL can not train a proper model for each task. As the training data decreases, our methods based on the variational Bayesian framework are able to better handle this challenging case by incorporating the shared knowledge into the prior of each task. The best and second-best results of average accuracy are respectively marked in bold and underlined.

C.2 Effectiveness of Variational Bayesian Approximation

Comparative results on performance of Bayesian approximation for representations $\mathbf{z}$ and classifiers $\mathbf{w}$ on the *Office-Home*, *Office-Caltech* and *ImageCLEF* datasets are shown in Tables C.7, C.8 and C.9, respectively. Both variational Bayesian representations and classifiers can benefit performance. Our method jointly infers the posteriors over feature representations and classifiers in a Bayesian framework and outperforms its variants on three benchmarks for most of train-test splits, which demonstrates the benefits of applying Bayesian inference to both classifiers and representations.

C.3 Effectiveness of Gumbel-Softmax Priors

The performance comparison of the proposed VMTL with different priors on the *Office-Home*, *Office-Caltech* and *ImageCLEF* datasets are shown in Tables C.10, C.11 and C.12, respectively. These approximated priors are obtained by combining posteriors of related tasks. "Mean" denotes that the prior of the current task is the mean of variational posteriors of other related tasks. "Learnable weighted" denotes that weights of mixing the variational posteriors of other related tasks are learnable. Our proposed prior by the Gumbel-Softmax technique to learn the mixing weights introduces uncertainty to the relationships among tasks to explore sufficient transferable information from other

Table C.7. Detailed results on performance of Bayesian approximation for representation **z** and classifier **w** on *Office-Home*.

z	**w**	5%					10%					20%				
		A	C	P	R	Avg.	A	C	P	R	Avg.	A	C	P	R	Avg.
×	×	37.6±0.4	31.5±0.3	68.5±0.2	63.8±0.2	50.4±0.1	51.0±0.2	41.6±0.1	76.0±0.3	69.2±0.3	59.5±0.1	56.6±0.3	51.8±0.5	80.9±0.3	72.9±0.4	65.6±0.1
✓	×	52.0±0.4	37.6±0.3	71.8±0.3	68.3±0.1	57.4±0.0	59.2±0.4	47.1±0.2	77.6±0.1	73.7±0.4	64.4±0.1	63.3±0.4	52.7±0.2	82.1±0.1	76.0±0.3	68.5±0.1
×	✓	51.6±0.3	37.8±0.1	70.7±0.4	66.9±0.2	56.8±0.1	57.6±0.3	47.2±0.2	77.4±0.3	72.5±0.1	63.7±0.1	62.5±0.4	53.5±0.3	82.0±0.2	76.0±0.3	68.5±0.1
✓	✓	53.5±0.1	39.2±0.2	71.5±0.3	69.0±0.1	**58.3±0.1**	60.2±0.2	47.5±0.2	78.2±0.3	74.0±0.2	**65.0±0.0**	64.0±0.3	53.5±0.4	82.5±0.2	76.8±0.1	**69.2±0.2**

Table C.8. Detailed results on performance of Bayesian approximation for representation **z** and classifier **w** on *Office-Caltech*.

z	**w**	5%					10%					20%				
		A	W	D	C	Avg.	A	W	D	C	Avg.	A	W	D	C	Avg.
×	×	90.0±0.7	89.4±0.8	95.0±1.1	83.5±0.5	89.5±0.3	93.6±0.1	97.0±0.6	92.1±0.7	86.3±0.4	92.3±0.2	95.0±0.2	94.5±0.7	96.0±1.2	86.8±0.2	93.1±0.1
✓	×	93.3±0.4	95.0±0.5	96.1±0.3	90.0±0.3	93.6±0.2	95.3±0.1	97.4±0.3	97.9±0.0	90.4±0.3	95.2±0.0	95.6±0.1	96.6±0.5	98.4±0.4	90.3±0.6	95.2±0.2
×	✓	93.2±0.1	95.5±0.3	95.7±0.5	89.6±0.2	93.5±0.1	94.8±0.2	97.3±0.4	97.8±0.4	90.6±0.3	95.1±0.1	95.6±0.2	96.6±0.3	99.2±0.3	89.8±0.4	**95.3±0.2**
✓	✓	93.7±0.2	95.2±0.3	96.4±0.4	89.7±0.4	**93.8±0.1**	95.4±0.1	97.6±0.1	97.4±0.4	90.9±0.2	**95.3±0.0**	95.6±0.2	95.6±0.4	98.4±0.5	91.1±0.3	95.2±0.1

Table C.9. Detailed results on performance of Bayesian approximation for representation **z** and classifier **w** on *ImageCLEF*.

z	**w**	5%					10%					20%				
		C	I	P	B	Avg.	C	I	P	B	Avg.	C	I	P	B	Avg.
×	×	88.3±0.5	73.2±0.5	61.2±0.9	40.0±0.8	65.7±0.4	90.6±0.8	79.3±0.2	66.3±0.5	51.9±0.6	72.0±0.3	92.9±0.5	86.5±0.2	71.9±0.8	56.0±0.8	76.8±0.3
✓	×	90.4±0.2	81.9±0.7	70.9±0.5	57.9±0.6	75.3±0.4	93.7±0.2	85.7±0.6	71.7±0.2	59.1±0.3	77.5±0.0	93.5±0.2	89.8±0.6	77.3±0.5	59.0±0.4	79.9±0.1
×	✓	91.4±1.1	81.9±0.5	71.8±0.6	58.4±0.5	75.9±0.2	93.0±0.6	86.3±0.4	72.0±0.6	60.0±0.6	77.8±0.3	93.8±0.5	88.1±0.5	77.1±0.8	59.0±0.4	79.5±0.2
✓	✓	91.5±0.2	83.5±0.6	71.6±0.8	58.2±0.7	**76.2±0.3**	94.2±0.2	86.0±0.5	71.7±0.6	59.8±0.4	**77.9±0.2**	93.9±0.1	89.4±0.2	78.0±0.4	59.5±0.4	**80.2±0.1**

Table C.10. Detailed results on performance of VMTL with different priors on *Office-Home*.

Priors	5%					10%					20%				
	A	C	P	R	Avg.	A	C	P	R	Avg.	A	C	P	R	Avg.
Mean	52.2±0.4	38.0±0.3	71.3±0.3	68.3±0.1	57.5±0.0	59.2±0.4	47.1±0.2	77.6±0.1	73.7±0.4	64.4±0.1	63.3±0.4	52.7±0.2	82.1±0.1	76.0±0.3	68.5±0.1
Learnable weighted	51.8±0.5	38.0±0.3	70.9±0.5	67.9±0.4	57.2±0.2	59.2±0.6	46.9±0.4	77.6±0.3	73.2±0.3	64.2±0.2	63.9±0.5	53.2±0.4	82.1±0.4	76.3±0.2	68.9±0.2
Gumbel-Softmax	53.5±0.1	39.2±0.2	71.5±0.3	69.0±0.1	**58.3±0.1**	60.2±0.2	47.5±0.2	78.2±0.3	74.0±0.2	**65.0±0.0**	64.0±0.3	53.5±0.4	82.5±0.2	76.8±0.1	**69.2±0.2**

Table C.11. Detailed results on performance of VMTL with different priors on *Office-Caltech*.

Priors	5%					10%					20%				
	A	W	D	C	Avg.	A	W	D	C	Avg.	A	W	D	C	Avg.
Mean	93.3±0.4	95.0±0.5	96.1±0.3	90.0±0.3	93.6±0.2	95.1±0.1	97.4±0.3	97.9±0.0	90.7±0.3	**95.3±0.0**	95.6±0.1	96.1±0.5	98.7±0.4	91.0±0.6	**95.4±0.2**
Learnable weighted	93.6±0.3	95.0±0.2	96.7±0.7	89.4±0.7	93.7±0.2	94.9±0.3	97.3±0.4	97.9±0.0	90.4±0.3	95.1±0.1	95.6±0.2	95.9±0.3	98.2±0.8	90.9±0.6	95.1±0.1
Gumbel-Softmax	93.7±0.2	95.2±0.3	96.4±0.4	89.7±0.4	**93.8±0.1**	95.4±0.1	97.6±0.1	97.4±0.4	90.9±0.2	**95.3±0.0**	95.6±0.2	95.6±0.4	98.4±0.5	91.1±0.3	95.2±0.1

Table C.12. Detailed results on performance of VMTL with different priors on *ImageCLEF*.

Priors	5%					10%					20%				
	C	I	P	B	Avg.	C	I	P	B	Avg.	C	I	P	B	Avg.
Mean	90.4±0.2	82.7±0.7	71.±0.5	57.8±0.6	75.5±0.4	93.4±0.2	86.4±0.6	71.5±0.2	59.1±0.3	77.6±0.0	93.6±0.2	89.7±0.6	77.8±0.5	59.4±0.4	80.1±0.1
Learnable weighted	90.1±0.4	82.1±0.2	71.2±1.0	58.5±0.6	75.5±0.4	93.4±0.3	85.6±0.5	71.5±0.5	59.1±0.6	77.4±0.2	93.5±0.3	89.5±0.2	77.9±0.7	59.6±0.4	80.1±0.2
Gumbel-Softmax	91.5±0.2	83.5±0.6	71.6±0.8	58.2±0.7	**76.2±0.3**	94.2±0.2	86.0±0.5	71.7±0.6	59.8±0.4	**77.9±0.2**	93.9±0.1	89.4±0.2	78.0±0.4	59.5±0.4	**80.2±0.1**

Table C.13. Performance comparison of different methods on the large-scaled dataset *DomainNet* for multiple tasks: Clipart (C), Infograph (I), Painting (P), Quickdraw (Q), Real (R) and Sketch (S).

Methods	4%						
	C	I	P	Q	R	S	Avg.
STL	23.0±0.2	7.1±0.3	30.4±0.1	5.2±0.2	58.7±0.3	16.3±0.3	23.5±0.2
VSTL	33.8±0.1	12.3±0.2	37.1±0.1	23.7±0.3	65.3±0.4	23.2±0.2	32.6±0.1
Bakker et al. [3]	28.3±0.4	10.7±0.2	35.4±0.1	22.4±0.3	59.9±0.3	21.7±0.3	29.7±0.3
Long et al. [33]	28.7±0.1	13.2±0.3	38.7±0.4	7.5±0.3	62.9±0.2	20.6±0.3	28.6±0.2
Kendall et al. [25]	30.2±0.3	11.6±0.4	37.3±0.4	29.7±0.2	62.1±0.5	22.2±0.3	32.2±0.3
Qian et al. [38]	32.5±0.3	12.6±0.2	40.4±0.4	25.8±0.5	64.4±0.2	25.4±0.3	33.5±0.3
BMTL	34.0±0.1	11.9±0.3	36.8±0.1	24.7±0.2	64.9±0.2	23.1±0.3	32.6±0.1
VBMTL	33.1±0.1	12.0±0.2	37.0±0.2	19.7±0.1	64.5±0.1	22.8±0.3	31.5±0.1
VMTL-AC	31.4±0.1	11.1±0.1	35.3±0.1	15.8±0.1	61.5±0.1	21.8±0.2	29.5±0.1
VMTL	36.4±0.2	14.8±0.2	40.0±0.1	19.0±0.1	65.5±0.1	26.1±0.1	**33.6±0.1**

Table C.14. Impact of L and M on performance. Experiments are conducted on *Office-Home* with a 5% train-test split.

L (=M)	1	10	20	30	40	50	60	70	80	90	100
VMTL-AC	56.1	**56.3**	56.1	56.2	56.3	56.1	**56.3**	56.0	56.2	**56.3**	55.8
VMTL	58.1	**58.3**	58.2	58.2	58.2	**58.3**	58.0	**58.3**	58.2	58.2	58.2

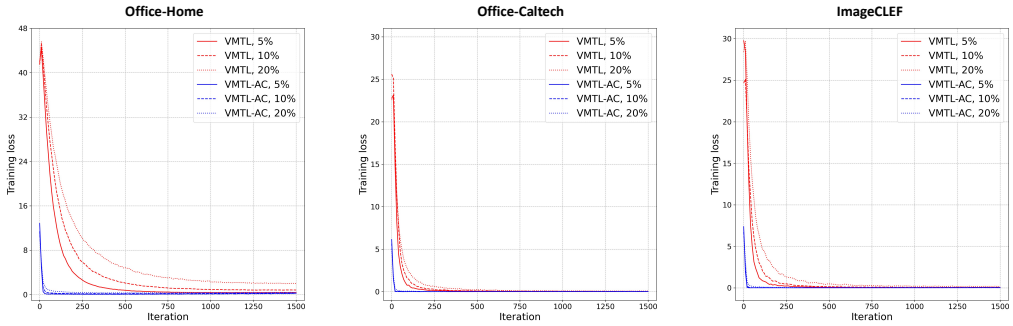

Fig. C.1. Illustration of training loss with iterations on *Office-Home*, *Office-Caltech* and *ImageCLEF*. VMTL-AC converges faster than VMTL under 5%, 10% and 20% train-test splits, which demonstrates the computational benefit of amortized learning.

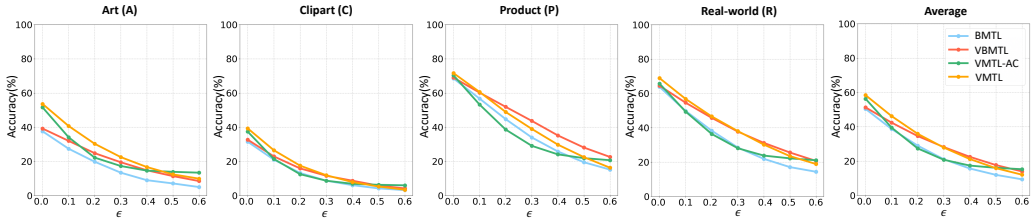

Fig. C.2. The performance for each task under different noise levels on the *Office-Home* dataset.

tasks. In the three datasets, our designed priors outperform other methods under most of the train-test splits. In addition, the results of *DomainNet* under the 4% train-test split are provided in Table C.13.

We learn different weights for $\mathbf{w}$ and $\mathbf{z}$, motivated by the fact that latent variables are different in terms of capturing uncertainty: $\mathbf{w}$ is at the category level while $\mathbf{z}$ at the instance level. To validate, we conduct experiments using the same Gumbel-Softmax weights for both variables. As shown in Table C.15, using different weights are slightly better than using the same Gumbel-Softmax weights.

Table C.15. Performance of our priors using the same weights for $\mathbf{w}$ and $\mathbf{z}$ or not on *Office-Home*.

Train-test split	5%	10%	20%
Same weights	58.2±0.1	64.8±0.1	68.7±0.3
Different weights	**58.3±0.1**	**65.0±0.0**	**69.2±0.2**

C.4 Sensitivity of the Hyper-parameter L and M

In this paper, L and M are the number of Monte Carlo samples for the variational posteriors of latent representations and classifiers, respectively. We ablate the sensitivity of L and M in Table C.14 on the *Office-Home* dataset. In practice, L and M are set to ten, which offer a good balance between accuracy and efficiency.

C.5 Fast Convergence of Amortized Classifiers

The computational advantage of amortized classifiers can be illustrated by the training loss as function of iteration on *Office-Home*, *Office-Caltech* and *ImageCLEF*. As shown in Fig. C.1, VMTL-AC converges faster than VMTL under 5%, 10% and 20% train-test splits, which demonstrates the computational benefit of amortized learning.

C.6 Robustness of Our Methods

We conduct experiments on the *Office-home* dataset to show the robustness of our methods against adversarial attacks. In our experiments, the adversarial attack is implemented by the fast gradient sign method [**?**] where ϵ denotes the noise level. As shown in Fig. C.2, under different noise levels, the proposed model VMTL outperforms BMTL. As the noise level increases, the proposed model VMTL-AC is more robust than other models.

C.7 A New Metric for Evaluating the Uncertainty Prediction

For Bayesian methods, it is necessary to quantify the model's ability of handling uncertainty. We looked into related references and didn't find such a measure for comparing the uncertainty prediction. Thus, we adopt a new metric for evaluating the uncertainty prediction, the ratio of the average entropy of failure cases and properly classified samples. If the ratio is higher, the Bayesian methods predict failure cases with more uncertainty and predict successful cases with more confidence. As shown in Table C.16, VMTL has higher entropy ratios, which demonstrates the effectiveness of our model to handle the uncertainty.

Table C.16. Entropy ratio (the higher the better) on *Office-Home*.

Train-test split	5%	10%	20%
Bakker et al.[3]	2.469	2.625	3.031
VBMTL	4.111	4.430	5.460
VMTL	**4.546**	**4.472**	**5.584**

C.8 Runtime Impact of the Sampling Steps

To investigate the runtime impact of the additional sampling steps we compare the actual training and inference time of the proposed method with that of deterministic approaches. As shown in Table C.17, compared to the deterministic baseline (BMTL), the training and inference time of our method increases as the number of MC samples is set higher. In this paper, the number of MC samples is set to be 10, which is computationally efficient while yielding good performance (Table C.14). In this case, our method does cost extra at training time but with 0.122s per iteration, this is still acceptable. When testing 1000 samples, our method only increases by an extra 10% test time of BMTL. Thus, our model doesn't cost much more time to surpass BMTL by 7.9% in terms of accuracy.

Table C.17. Runtime impact (seconds) of sampling step on Office-Home with 5% split.

Methods	BMTL	VMTL			
MC samples	-	1	10	50	100
Training (per iteration)	0.040	0.098	0.122	0.197	0.320
Inference (per 1000 test samples)	0.325	0.343	0.357	0.371	0.426

References

[1] C. Archambeau, S. Guo, and O. Zoeter. Sparse bayesian multi-task learning. In *NIPS*, volume 1, page 41, 2011.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in neural information processing systems*, pages 41–48, 2007.

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

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

[5] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015.

[6] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[7] F. J. Bragman, R. Tanno, S. Ourselin, D. C. Alexander, and J. Cardoso. Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution kernels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1394, 2019.

[8] R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[9] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR, 2018.

[10] C. Cremer, X. Li, and D. Duvenaud. Inference suboptimality in variational autoencoders. *arXiv preprint arXiv:1801.03558*, 2018.

[11] H. Daumé III. Bayesian multitask learning with latent hierarchies. *arXiv preprint arXiv:0907.0783*, 2009.

[12] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[13] C. Finn, K. Xu, and S. Levine. Probabilistic model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9537–9548, 2018.

[14] Y. Gao, H. Bai, Z. Jie, J. Ma, K. Jia, and W. Liu. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11543–11552, 2020.

[15] S. Gershman and N. Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014.

[16] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE, 2012.

[17] J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, and R. E. Turner. Meta-learning probabilistic inference for prediction. *arXiv preprint arXiv:1805.09921*, 2018.

[18] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.

[19] P. Guo, C.-Y. Lee, and D. Ulbricht. Learning to branch for multi-task learning. In *International Conference on Machine Learning*, pages 3854–3863. PMLR, 2020.

[20] T. Heskes. Empirical bayes for learning to learn. 2000.

[21] A. Jalali, S. Sanghavi, C. Ruan, and P. K. Ravikumar. A dirty model for multi-task learning. In *Advances in neural information processing systems*, pages 964–972, 2010.

[22] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[23] P. K. Jawanpuria, M. Lapin, M. Hein, and B. Schiele. Efficient output kernel learning for multiple tasks. In *Advances in neural information processing systems*, pages 1189–1197, 2015.

[24] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *ICML*, volume 2, page 4, 2011.

[25] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[26] H. Kim, A. Mnih, J. Schwarz, M. Garnelo, A. Eslami, D. Rosenbaum, O. Vinyals, and Y. W. Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

[27] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[30] H. Liu, M. Palatucci, and J. Zhang. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 649–656, 2009.

[31] L. Liu, Y. Li, Z. Kuang, J.-H. Xue, Y. Chen, W. Yang, Q. Liao, and W. Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021.

[32] S. Liu, E. Johns, and A. J. Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019.

[33] M. Long, Z. Cao, J. Wang, and S. Y. Philip. Learning multiple tasks with multilinear relationship networks. In *Advances in neural information processing systems*, pages 1594–1603, 2017.

[34] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[35] E. Nalisnick and P. Smyth. Learning priors for invariance. In *International Conference on Artificial Intelligence and Statistics*, pages 366–375. PMLR, 2018.

[36] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

[37] T. K. Pong, P. Tseng, S. Ji, and J. Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20(6):3465–3489, 2010.

[38] W. Qian, B. Chen, and F. Gechter. Multi-task variational information bottleneck. *arXiv preprint arXiv:2007.00339*, 2020.

[39] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[40] Y. Shi, N. Siddharth, B. Paige, and P. H. Torr. Variational mixture-of-experts autoencoders for multi-modal deep generative models. *Advances in Neural Information Processing Systems*, 2019.

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[42] G. Strezoski, N. v. Noord, and M. Worring. Many task learning with task routing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1375–1384, 2019.

[43] X. Sun, R. Panda, R. Feris, and K. Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *arXiv preprint arXiv:1911.12423*, 2019.

[44] H. Takahashi, T. Iwata, Y. Yamanaka, M. Yamada, and S. Yagi. Variational autoencoder with implicit optimal priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5066–5073, 2019.

[45] M. K. Titsias and M. Lázaro-Gredilla. Spike and slab variational inference for multi-task and multiple kernel learning. In *Advances in neural information processing systems*, pages 2339–2347, 2011.

[46] J. Tomczak and M. Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223, 2018.

[47] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *(IEEE) Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[48] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(Jan):35–63, 2007.

[49] K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on Machine learning*, pages 1012–1019, 2005.

[50] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.

[51] Y. Zhang and Q. Yang. Learning sparse task relations in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[52] Y. Zhang and Q. Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[53] Y. Zhang and D. Yeung. Multi-task learning using generalized t process. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 964–971, 2010.

[54] Y. Zhang and D.-Y. Yeung. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*, 2012.

[55] Y. Zhang, Y. Zhang, and W. Wang. Deep multi-task learning via generalized tensor trace norm. *arXiv preprint arXiv:2002.04799*, 2020.