SHSML: A Stochastic Approach to Hierarchically Structured Meta-Learning for Improved Inference and Confidence

Zhuoran Li[1], Xuefeng Chen[1], Liang Feng[1,†], Zhou Wu[2], and Xin Xu[3]
[1]Colledge of Computer Science, Chongqing University, Chongqing, China
[2]School of Automation, Chongqing University, Chongqing, China
[3]School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China

***Abstract*—Meta-learning aims to train models that can learn from a variety of related tasks and use that learned knowledge to solve new and unseen tasks more efficiently. However, it often suffers from handling a sequence of tasks originated from different distributions. To address this issue, the hierarchically structured meta-learning (HSML) has been proposed in the literature. The HSML utilizes a hierarchically structured cluster to address task relationship and similarity, which is regarded as transferable knowledge. Despite the success enjoyed by the HSML, it is worth noting that task uncertainty is ignored by HSML in inference on new tasks by point-estimate manner, which could lead to overconfidence on inappropriate inference results. Taking this cue, in this paper, we propose a stochastic HSML (SHSML) algorithm, which extends HSML with uncertainty awareness by representing each task-specific model as a stochastic variable. By sampling multiple task-specific models and ensembling their inference results instead of point-estimation of HSML, the SHSML is able to mitigate the overconfidence problem in HSML and gives a confidence range of inference. To evaluate the performance of the proposed approach, comprehensive empirical studies are conducted on common curve regression task against state-of-the-art meta-learning algorithms. The obtained results confirmed the efficacy of the proposed approach in handling both task uncertainty and heterogeneity in meta-learning.**

***Index Terms*—Meta learning, Knowledge transfer, Hierarchical structure, Uncertainty**

I. Introduction

Meta-learning aims to enable efficient and rapid adaptation to new tasks with minimal training samples. One implementation is acquiring knowledge from multiple interrelated tasks and applying it to novel tasks [1] or splitting tasks through feature selection [2] and learning knowledge through select features. The learning process on these related tasks reveals common insights across diverse tasks originating from the same distribution [3]. Recent achievements in meta-learning, particularly in domains such as few-shot robotic control [4], object recognition and detection [5], as well as classification [6], have generated significant interest and enthusiasm within the research community. With the great success of machine learning in real-world applications [7], the importance of using meta-learning to design efficient and robust machine learning algorithms becomes increasingly prominent.

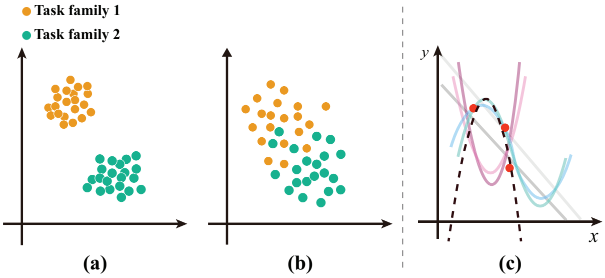


Fig. 1. An illustration of task uncertainty. (a) Task set sampled from two families with low uncertainty. (b) Task set sampled from two families with high uncertainty. (c) Uncertainty caused by tiny dataset with only 3 examples with noise. These 3 points are sampled from a quadratic curve shown as dashed black line. But due to uncertainty introduced by limited number of samples and sampling noise, these 3 points can be considered to fit other kinds of curves in a heterogeneity scenario.

Despite the recent achievements across various domains, meta-learning approaches still face two significant challenges: the task heterogeneity challenge and the task uncertainty challenge. The task heterogeneity challenge stems from a critical assumption made by most existing meta-learning algorithms, which assumes that all tasks encountered during meta-training and meta-testing originate from the same distribution [3], [8], [9]. However, this assumption may not hold in many complex real-world scenarios. Consequently, meta-learning algorithms struggle when dealing with tasks derived from different distributions. These heterogeneity-sampled tasks are not compatible with the globally shared knowledge meta learned by meta-learning algorithms and may lead to a degradation in performance. To address this issue, one notable algorithm proposed in the literature is HSML (Hierarchically Structured Meta-Learning) [10], which introduces a robust mechanism to extract transferable domain knowledge from task datasets. HSML extracts and transfers knowledge between tasks by learning task similarity and relationships, which is proofed to be efficiency in previous works [11]. HSML utilizes a hierarchically structured cluster to address task similarity and relationships, extracting transferable knowledge from related task clusters. This transferable knowledge is then employed for

[†] L. Feng (Liangf@cqu.edu.cn) is the corresponding authors.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00231

knowledge customization, tailoring task-specific parameters based on meta-parameters conditioned on transfer knowledge. This mechanism maximizes the utilization of domain knowledge in the task space, surpassing the performance of many state-of-the-art meta-learning methods. However, a noteworthy limitation of HSML lies in its inference process, which is designed in a point-estimated manner, neglecting the uncertainty introduced by small datasets. The tiny dataset might not provide sufficient information for HSML to accurately estimate task relationships and extract transferable knowledge with high uncertainty using a point-estimated approach.

Another challenge encountered by meta-learning algorithms is the task uncertainty challenge. This challenge arises from the inherent uncertainty in tasks, primarily caused by the limited dataset size associated with each task. As depicted in Fig. 1, learning an appropriate model becomes challenging when the dataset comprises a small number of noisy samples. To address this issue, several approaches have been proposed in the field of meta-learning [12], [13], [13]–[17]. These approaches often extend the Model-Agnostic Meta-Learning (MAML) framework to a probabilistic form, where the meta-learned model initialization or task-specific model is treated as a stochastic variable sampled from a learned probability distribution. For instance, ST-MAML [17], considers task representation as a stochastic variable drawn from a Gaussian distribution with learnable parameters, known as stochastic task representation. Multiple task-specific model parameters can then be customized from meta-parameters conditioned on samples of the stochastic task representation. ST-MAML leverages this approach to latently model task-specific parameters as stochastic variables, effectively mitigating uncertainty. The inference results of these task-specific models can be ensemble-averaged to alleviate uncertainty and provide a confidence range for predictions. Importantly, as HSML also employs a parameter gate to customize meta-parameters, it is straightforward to further extend HSML with stochastic task representation, showcasing the potential for seamless integration of uncertainty-mitigating techniques.

Keeping the above in mind, in this paper, we present an extension to HSML called Stochastic Hierarchical Stochastic Meta-learning (SHSML), which incorporates uncertainty awareness. Our approach transforms point-estimated transferable knowledge into a stochastic variable, allowing us to tailor multiple task-specific model parameters from meta-parameters using samples of this stochastic transferable knowledge. As a result, the task-specific models generated can be synergistically combined, leading to more robust and accurate inferences, while also providing a confident range for the inference results. To validate the effectiveness of our method, we conduct empirical experiments focusing on curve regression. Our approach is comprehensively evaluated through comparisons with related algorithms, including MAML [3], HSML [10], MetaSGD [9], and Vampire [13] on multiple curve regression tasks, according to [10].

In summary, our work makes the following contributions:

- We extend HSML with the capability to handle uncertainty and propose SHSML, which effectively address uncertainty arising from limited datasets and task heterogeneity.
- The proposed SHSML effectively tackles uncertainty arising from limited datasets and task heterogeneity.
- Experimental results on multi-modal curve regression task demonstrate that SHSML outperforms sseveral state-of-the-art meta-learning methods in mitigating uncertainty when dealing with small datasets with noise and diverse tasks.

II. Preliminary

A. Task Heterogeneity Challenge in Meta-Learning

Traditional meta-learning algorithms encounter challenges when dealing with tasks derived from different distributions, which is known as the heterogeneity task setup. n this setup, tasks are represented as$\mathcal{T}_i \sim p(\mathcal{T}) \in \mathcal{E}$, where $\mathcal{T}$ denotes sampled tasks and $\mathcal{E} = \{p_1(\mathcal{T}), p_2(\mathcal{T}), \cdots\}$ represents the task environment. To improve the performance of meta-learning in the presence of heterogeneous tasks, various approaches have been proposed in the literature.

For instance, MUMOMAML incorporates network tailoring by generating task-specific models from meta-models through task embedding. TSA-MAML [18] conducts a theoretical analysis of MAML and utilizes model parameters for each task to measure task similarity learned by vanilla MAML. Moreover, TAML [19] introduces an entropy-based method for unbiased parameter initialization. HSML [10] employs a hierarchical structure to extract task similarity and transferable knowledge, which plays a crucial role in model customization. ARML [20] integrates a knowledge graph as a manager of meta-knowledge from diverse tasks, facilitating the retrieval of relevant knowledge for model tailoring. The idea of transferring meta-knowledge between heterogeneous tasks has also been utilized to enhance the efficiency of multitask evolutionary algorithms [21]–[23].

B. Task Uncertainty in Meta-Learning

As shown in Fig. 1, the inherent uncertainty in a meta-learning scenario is exacerbated by the limited support set for each task and the introduction of heterogeneity through task sampling. In the literature, probabilistic methods have been employed within meta-learning to tackle this challenge. For example, LLAMA [16] enhances the robustness of learned models by incorporating a Gaussian distribution to model task-specific parameters. However, the computation of the Hessian matrix introduced by the Laplace method proves to be computationally expensive. In contrast, PLATIPUS [12] focuses on learning a distribution for meta-parameters and utilizes variational inference (VI) to alleviate the computational burden. To streamline the task adaptation process and work in conjunction with VI, BMAML [15] utilizes a closed-form solution based on Stein Variational Gradient Descent (SVGD) [24]. Additionally, MAHA [25] leverages a neural process to generate well-clustered and interpretable task representations.

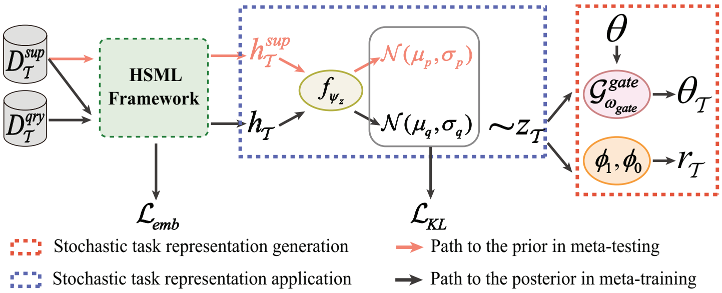

$D_{\mathcal{T}}^{sup}$
$D_{\mathcal{T}}^{qry}$
HSML Framework
$\mathcal{L}_{emb}$
$h_{\mathcal{T}}^{sup}$
$h_{\mathcal{T}}$
f_{ψ_z}
$\mathcal{N}(\mu_p,\sigma_p)$
$\mathcal{N}(\mu_q,\sigma_q)$
$\sim z_{\mathcal{T}}$
$\mathcal{L}_{KL}$
θ
$\mathcal{G}_{\omega_{gate}}^{gate}$
$\theta_{\mathcal{T}}$
ϕ_1,ϕ_0
$r_{\mathcal{T}}$
Stochastic task representation generation
Stochastic task representation application
Path to the prior in meta-testing
Path to the posterior in meta-training

Fig. 2. The framework of the proposed SHSML. In SHSML, the transferable knowledge $h_{\mathcal{T}}$ is generated using the original HSML framework. It then be utilized to generate stochastic task representation $z_{\mathcal{T}}$. Unlike HSML which involves a single forward path in the inner-loop, our SHSML generate two transferable knowledge $h_{\mathcal{T}}^{sup}$ and $h_{\mathcal{T}}$ on both the support set and the complete dataset (including the support set and query set) during the training phase (black arrows). These two transfer knowledge are used to approximate the prior and posterior distributions of the stochastic task representation, which are used to compute the KL-divergence $\mathcal{L}_{KL}$. Additionally the task embedding loss $\mathcal{L}_{emb}$ is computed on the complete dataset. Due to the unavailability of labels in meta-testing phase, SHSML only computes one transferable knowledge representation on the support set.

Similarly, Vampire [13] concentrates on learning a prior distribution of meta-parameters, while efficiently deriving the posterior distribution of task-specific parameters through gradient descent.

Our proposed SHSML also draws inspiration from a similar foundation. In our approach, we tailor the task-specific model from the meta-parameter by a stochastic variable drawn from Gaussian distribution. The following section will provide the details of our proposed SHSML.

III. Methodology

In this section, we present the details of the proposed SHSML algorithm. SHSML addresses uncertainty in a task heterogeneous setup by introducing a stochastic task representation to the HSML workflow. It models task-specific parameters as a stochastic variable sampled from a learnable distribution. To mitigate the uncertainty, the inference results of multiple task-specific models are ensembled as the final output. As shown in Fig. 2, the crucial aspect of SHSML is the mechanism for stochastic task representation generation (STR) and the utilization of STR in model tailoring. The subsequent sections will delve into the specifics of the stochastic task representation and outline the training methodology employed by SHSML.

A. Stochastic Task Representation

Similar to traditional meta-learning algorithms, the hierarchically structured cluster in HSML may suffer from performance degradation in the presence of uncertainty. This can lead to the provision of inappropriate transferable knowledge, which hinders effective model tailoring. To address this issue, we introduce a stochastic variable named stochastic task representation. Unlike a fixed value for transferable knowledge in HSML, multiple samples of the stochastic task representation for a given task aid in mitigating overconfidence. The acquisition of the stochastic task representation is formulated in (1).

$$\begin{aligned} p(z_{\mathcal{T}}|\mathcal{T}) &= \mathcal{N}(\mu,\sigma) \\ \mu,\sigma &= f_{\psi_z}(h_{\mathcal{T}}) \end{aligned} \tag{1}$$

where we model the distribution of stochastic task representation as a Gaussian distribution. A multi-layer perceptron (MLP) f_{ψ_z} parameterized by ψ_z determines μ and σ of the Gaussian distribution, taking transferable knowledge $h_{\mathcal{T}}$ as input. Multiple samples of stochastic task representation $z_{\mathcal{T}}$ are drawn from $p(z_{\mathcal{T}}|\mathcal{T})$ and contribute to model tailoring, generating multiple task-specific models.

B. Knowledge Adaptation

The task-specific parameter $\theta_{\mathcal{T}}$ for task $\mathcal{T}$ is tailored from the meta-parameter using the stochastic task representation $z_{\mathcal{T}}$. Initially, a parameter gate is applied to the meta-parameter as described by (2)

$$\theta_{\mathcal{T}} = G_{\omega_{gate}}^{gate}(\theta) = [\theta_b; \sigma(W z_{\mathcal{T}} + b) \odot \theta_c] \tag{2}$$

The parameter gate $G_{\omega_{gate}}^{gate}$ is parametered by ω_{gate}, which including learnable parameter W and b. The parameter gate performs a linear transformation on stochastic task representation $z_{\mathcal{T}}$, followed by a sigmoid function $\sigma(\cdot)$. The output of sigmoid function $\sigma(\cdot))$ serves as a weight vector that is used to tailor custom parameter θ_c through element-wise production $\odot$. The tailored θ_c is concatenated with base parameter θ_b to form the complete tailored task-specific parameter $\theta_{\mathcal{T}}$. Typically, the custom parameter represents the final fully-connection layer of a Convolutional Neural Network (CNN) or the last layer of a MLP, while the unchanged parts are referred to as the base parameter θ_b. This approach employed by ST-MAML [17] can effectively reduce the number of learnable parameters in the parameter gate. Additionally, a fully connected layer transforms $z_{\mathcal{T}}$ into augmented feature representations denoted as $r_{\mathcal{T}}$. These feature representations will be concatenated with input of each data sample $x_{\mathcal{T}} = \{x_1, x_2, \cdots, x_n\}$ to introduce clustered stochasticity into model inputs, as shown in (4)

Algorithm 1 Stochastic Hierarchical Structured Meta Learning

```
Input: $\mathcal{E}$: task environment; $L$, $\{K^1, \cdots, K^L\}$: number of layers and clusters in each layer of hierarchically structured clustering; $\alpha$, $\beta$: step sizes for outer loop and inner loop; $\gamma, \eta$: scaling factors. $m$: number of samples of stochastic task representation; $C$: number of update step in inner loop
Output: Meta parameters $\Theta$ and loss on query dataset of given test task $\mathcal{L}_{in}^{qry}$
{Phase 1: Meta training}
Initialize $\Theta$
while criterion is not satisfied do
    $\mathcal{L}_{meta} \leftarrow 0$
    Sample a batch of tasks $\mathcal{T} \sim \mathcal{E}$
    for all sampled $\mathcal{T}$ do
        Sample $D_{\mathcal{T}}^{sup}$, $D_{\mathcal{T}}^{qry}$ from $\mathcal{T}$
        $h_{\mathcal{T}}, \mathcal{L}_{emb} \leftarrow HSML(\mathcal{T}, L, K_1, \cdots, K_L)$
        $\mu_p, \sigma_p \leftarrow f_{\psi_z}(h_{\mathcal{T}})$, $\mu_q, \sigma_q \leftarrow f_{\psi_z}(h_{\mathcal{T}})$
        $\mathcal{L}_{KL} \leftarrow KL(\mathcal{N}(\mu_p, \sigma_p)|\mathcal{N}(\mu_q, \sigma_q))$
        $\mathcal{L}_{in}^{qry} \leftarrow InnerLoop(\mathcal{D}_{\mathcal{T}}^{sup}, \mathcal{D}_{\mathcal{T}}^{qry}, h_{\mathcal{T}}, \mu_q, \sigma_q, m, C, \beta)$
        $\mathcal{L}_{meta} \leftarrow \mathcal{L}_{meta} + \gamma(\mathcal{L}_{emb} - \mathcal{L}_{ELBO})$
    end for
    $\Theta \leftarrow \Theta - \alpha \nabla_{\Theta} \mathcal{L}_{meta}$
end while
return $\Theta$
{Phase 2: Meta testing}
$\mathcal{L}_{in}^{qry} \leftarrow 0$
for $i = 1 \rightarrow M$ do
    $z_{\mathcal{T}} \sim \mathcal{N}(\mu, \sigma)$
    $\theta_{\mathcal{T}} \leftarrow [\theta_b; \sigma(Wz_{\mathcal{T}} + b) \odot \theta_c]$
    Compute augmented input $\hat{x}^{sup}$ and $\hat{x}^{qry}$ using (4)
    Update $\theta_{\mathcal{T}}$ and $r_{\mathcal{T}}$ through inner loop for $C$ times.
end for
$\mathcal{L}_{in}^{qry} \leftarrow \mathcal{L}_{in}^{qry} + \mathcal{L}_{in}(\hat{x}^{qry}, y^{qry}; \theta_{\mathcal{T}})$
return $\mathcal{L}_{in}^{qry}$
```

$$\hat{x_{\mathcal{T}}} = [x_{\mathcal{T}}; r_{\mathcal{T}}] = \{[x_1; r_{\mathcal{T}}], \cdots [x_n; r_{\mathcal{T}}]\} \tag{3}$$

$$r_{\mathcal{T}} = \phi_1 z_{\mathcal{T}} + \phi_0 \tag{4}$$

where ϕ_0 and ϕ_1 are learnable parameters used to compute $r_{\mathcal{T}}$ through linear transformation on stochastic task representation $z_{\mathcal{T}}$. The forward process of task-specific model for each data sample in task $\mathcal{T}$ is depicted in (5).

$$\hat{y}^i = f_{\theta_b}(f_{\theta_c}(\hat{x}_{\mathcal{T}}^i)) \tag{5}$$

Here $\hat{y}^i$ denotes predicted label for the corresponding augmented input $\hat{x}_{\mathcal{T}}^i$, f_{θ_b} and f_{θ_c} denotes the part of model corresponding to base parameter θ_b and the custom parameter θ_c, respectively. It is worth noting that, multiple samples of $z_{\mathcal{T}}$ can generate multiple task-specific models for a single task $\mathcal{T}$, and the outputs of these task-specific models are ensembled to mitigate uncertainty and provide an estimation of the confidence range.

C. Meta-training and Meta-testing

The entire process of meta-training and meta-testing of SHSML is outlined in Algorithm 1. The meta-training process of SHSML, presented in phase 1 of Algorithm 1, follows the HSML framework but introduces stochastic task representations in the inner-loop. Unlike HSML, the learnable parameters associated with the current task $\mathcal{T}$ in the inner-loop include tasks-specific model parameters $\theta_{\mathcal{T}}$ and input augmentation $r_{\mathcal{T}}$. In (7), these two learnable parameters are referred to as task-specific parameters and undergo several gradient-descent updates w.r.t empirical loss in (6) on the support set of the current task.

$$\mathcal{L}_{in}(\mathcal{T}) = \frac{1}{N^{sup}} \sum_{j=1}^{N^{sup}} \mathcal{L}(f_{\theta_{\mathcal{T}}}(\hat{x}_j^{sup}), y_j^{sup}) \tag{6}$$

where $f_{\theta_{\mathcal{T}}}$ represents the task-specific model parameterized by $\theta_{\mathcal{T}}$. The notation $\hat{x}_j^{sup}$ refers to the augmented input of the j-th data sample x_j^{sup} and y_j^{sup} is the label corresponding to x_j^{sup}. Additionally, N^{sup} is the size of support set.

$$\begin{aligned} r_{\mathcal{T}} &\leftarrow r_{\mathcal{T}} - \beta \frac{\partial \mathcal{L}_{in}(\mathcal{T})}{\partial r_{\mathcal{T}}} \\ \theta_{\mathcal{T}} &\leftarrow \theta_{\mathcal{T}} - \beta \frac{\partial \mathcal{L}_{in}(\mathcal{T})}{\partial \theta_{\mathcal{T}}} \end{aligned} \tag{7}$$

where β is the inner update step.

It is worth highlighting that multiple different stochastic task representations $z_{\mathcal{T}} \sim p(z_{\mathcal{T}}|\mathcal{T})$ are sampled in each inner loop, and the corresponding task-specific parameters are optimized individually. Finally, the validation losses of the different learnable parameter samples in the inner loop are accumulated to form the meta loss.

Since SHSML introduces a probabilistic method into HSML process, the original meta objective in HSML becomes intractable. Instead, we choose to maximize the evidence lower bound (a.k.a ELBO [26]) given by:

$$\mathcal{L}_{ELBO}(\mathcal{T}) = \mathop{\mathbb{E}}_{z_{\mathcal{T}} \sim q_{\mathcal{T}}} \mathcal{L}_{in}^{qry}(\mathcal{T}) - \eta \mathcal{L}_{KL}(\mathcal{T}) \tag{8}$$

where $q_{\mathcal{T}} = q(z_{\mathcal{T}}|\mathcal{T})$ represents the Gaussian distribution of stochastic task representation, and $\mathcal{L}_{KL}(\mathcal{T})$ denotes the KL-divergence between the prior $p(z_{\mathcal{T}}|\mathcal{D}_{\mathcal{T}}^{sup})$ and posterior $q(z_{\mathcal{T}}|\mathcal{T})$ of $z_{\mathcal{T}}$. To balance performance learning and posterior approximation in the KL-divergence term of ELBO, we introduce a factor η.

It is noteworthy that computing KL-divergence requires the prior and posterior of $z_{\mathcal{T}}$, which we approximate using Gaussian distribution derived from task datasets. Therefore we use both the support set and the query set to infer parameters for the prior, denoted as $\mathcal{N}(\mu_p, \sigma_p)$ and only the support set to derive the posterior $\mathcal{N}(\mu_q, \sigma_q)$. As a result, we compute the task embedding and transferable knowledge twice on different

TABLE I
REGRESSION MSE WITH 95% CONFIDENCE INTERVAL ON TOY REGRESSION TASK SET. ("+" AND − DENOTE BENCHMARK ALGORITHMS STATISTICALLY SIGNIFICANTLY BETTER AND WORSE THAN WE PROPOSED SHSML, RESPECTIVELY)

Model	MSE (±95% confidence interval)			
	5-shot		10-shot	
	$\sigma = 0.3$	$\sigma = 1.0$	$\sigma = 0.3$	$\sigma = 1.0$
MAML [3]	17.01 ± 3.16% −	18.26 ± 3.13% −	16.34 ± 4.12% −	17.58 ± 4.05% −
MetaSGD [9]	29.26 ± 8.24% −	32.76 ± 9.81% −	21.08 ± 12.97% −	18.52 ± 7.94% −
HSML [10]	1.60 ± 0.42% −	2.80 ± 0.51% −	0.56 ± 0.17% +	1.84 ± 0.29% +
Vampire [13]	19.98 ± 4.25% −	21.76 ± 4.12% −	32.90 ± 4.82% −	33.44 ± 4.28% −
SHSML(ours)	1.19 ± 0.56%	1.29 ± 0.69%	0.58 ± 0.41%	2.18 ± 0.56 %

datasets for computing different μ and σ in line 9 of Algorithm 1.

By employing the amortized variational technique, we can sample task-specific parameter initializations by first sampling $z_{\mathcal{T}}$ from approximated posterior $\mathcal{N}(\mu_q, \sigma_q)$. We then apply a deterministic transformation using (2) and (4) to obtain the task-specific parameters. The minimization objective of SHSML combines various objectives including accumulated query loss in the inner loop $\mathcal{L}_{in}^{qry}$, the ELBO objective, and the task embedding loss $\mathcal{L}_{emb}$ introduced by the HSML framework. The complete objective is formulated as shown in (9).

$$\mathcal{L}_{meta} = \mathbb{E}_{\mathcal{T} \sim \mathcal{E}}(\gamma \mathcal{L}_{emb} - \mathcal{L}_{ELBO}) \tag{9}$$

where γ is scaling factor of task embedding loss, and $\Theta = \{\theta, \psi_{img}, \psi_{enc}, \psi_{dec}, \psi_H, \psi_z, \omega_{gate}, \phi_0, \phi_1\}$. Phase 1 of Algorithm 1 outlines the complete process of meta training. In the outer-loop, we minimize the meta loss in (9) by considering a meta batch m tasks.

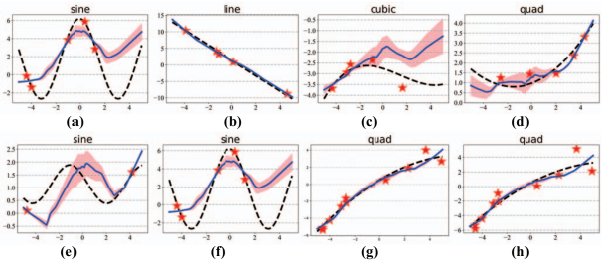
The meta-testing process is much similar to the inner-loop of SHSML shown in phase 2 of Algorithm 1. Differently, only the support set is accessible in meta-testing. So $z_{\mathcal{T}}$ is sampled from posterior $q(z_{\mathcal{T}} | \mathcal{D}_{\mathcal{T}}^{sup})$ which only relies on support set.

IV. EXPERIMENTS

In this section, we compare SHSML with several baseline meta-learning approaches to assess its efficiency. Our experiment setting is based on HSML [10], which includes several 1-D curve regression task families. Under this setting, the task heterogeneity is reflected in the variety of curve coefficients and families. Additionally, we achieve various levels of task uncertainty by manipulating the size of the support set in the few-shot learning scenario and injecting the Gaussian noise with different scales into data samples. The baseline methods we used are: (1) MAML [3], (2) MetaSGD [9], (3) Vampire [13] and (4) HSML [10]

A. Toy Regression

Dataset and Experimental Settings: In the toy regression problem, we consider a task environment $\mathcal{E}$ consisting of four different task families. In this paper, we adopt similar settings as HSML [10], where the underlying families are (1) *Sinusoid*: $y(x) = A\sin(\omega x) + b + \epsilon$, $A \sim U[0.1, 5.0]$, $\omega \in U[0, \pi]$; (2) *Line*: $y(x) = Ax + b + \epsilon$, $A \sim U[-3.0, 3.0]$, $b \sim U[-3.0, 3.0]$; (3)*quadratic*: $y(x) = Ax^2 + bx + c + \epsilon$, $A \sim U[-0.2, 0.2]$, $b \sim U[-3.0, 3.0]$; (4) *cubic*: $y(x) = Ax^3 + bx^2 + cx + d + \epsilon$, $A \sim U[-0.1, 0.1]$, $b \sim U[-0.2, 0.2]$, $c \sim U[-2.0, 2.0]$, $d \sim U[-3.0, 3.0]$. Here, $U[\cdot, \cdot]$ represents a uniform distribution, and ϵ represents a random noise signal sampled from a Gaussian distribution $\mathcal{N}(0, \sigma)$. Each task is randomly sampled from one of the four underlying functions, with input x uniformly sampled from $U[-5.0, 5.0]$ for both the meta-training and meta-testing tasks.



(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 3. Visualization of inference under different uncertainty settings. Red stars denote the training data, the black dot lines represent the ground truth, and the solid blue line with a light red shadow represents the inference mean and confidence range. In each sub-figure, the x-axis represents input $x_{\mathcal{T}}$ of each curve regression task and the y-axis denotes the curve value. (a-d): 5-shot training set with noise $\sigma = 0.3$. (e-f): constant noise level $\sigma = 0.3$ with different training set. $|\mathcal{D}_{\mathcal{T}}^{tr}| = 2$ for (e) and $|\mathcal{D}_{\mathcal{T}}^{tr}| = 5$ for (f). (g-h): 10-shot training set with various noise levels σ. Noise level $\sigma = 0.3$ for (g) and $\sigma = 1.0$ for (h)

All algorithms are applied on 5-shot and 10-shot settings separately, and evaluated using the mean square error (MSE). Our models adopt the same fully-connected architecture as HSML [10], which comprises 2 hidden layers, each consisting of 40 neurons. In the case of SHSML, the final layer responsible for generating the output is considered as the custom model θ_c. For hierarchically structured clustering, we adopt the settings from HSML, which consists of three layers with 4, 2, and 1 clusters in each layer, respectively.

Results of Regression Performance Each algorithm was trained on a diverse set of approximately 10000 tasks and subsequently evaluated on over 1000 new tasks. To assess the robustness of our method, we varied the number of available training examples (k-shot) and introduced different levels of noise to the data. The fitting curves for different scenarios are visualized in Fig. 3 and the evaluation results are summarized in Table. I.

As shown in Table. I, both HSML and SHSML outperformed the other baseline methods. Specifically, SHSML achieves the best result in the 5-shot settings and is runner-up in the 10-shot settings. In the 5-shot settings, the availability of less data compared to the 10-shot scenarios leads to a higher level of uncertainty. Our proposed SHSML achieves superior performance compared to other benchmarks by addressing uncertainty introduced by heterogeneity and limited datasets. In the 10-shot settings, HSML demonstrates better performance because the uncertainty caused by the few-shot dataset is reduced, and the can partially mitigate the uncertainty arising from heterogeneous problems.

Fig. 3 illustrates the predictions of SHSML on different curves with varying uncertainty. It can be observed that as tasks become more ambiguous due to limited training data or increased noise, the sampled solutions tend to cover a wider solution space and may exhibit similarities to other task families, leading to potential mistakes. These findings highlight the effectiveness of our proposed SHSML approach in addressing the uncertainty challenge in heterogeneous task settings. By leveraging hierarchical clustering and stochastic task representation, SHSML is capable of capturing and leveraging transferable knowledge across different task families, resulting in enhanced robustness as compared to other methods.

V. Conclusion

The challenges posed by task heterogeneity and task uncertainty are crucial in the field of meta-learning. Existing methods have primarily focused on addressing one challenge while overlooking the other. In this paper, we extend HSML by incorporating a stochastic task representation, which effectively mitigates uncertainty by learning a distribution of solutions for uncertain tasks. This approach provides a novel perspective on simultaneously tackling both task heterogeneity and task uncertainty. Through comprehensive empirical studies, we demonstrate the effectiveness of SHSML in learning from a diverse set of tasks characterized by both heterogeneity and uncertainty. In future work, we plan to enhance the mechanisms for learning transferable knowledge, aiming to achieve even more effective utilization of shared information across different tasks.

Acknowledgment

This work was supported by the National Key R&D Program of China (2022YFC3801700), China Postdoctoral Science Foundation (2022M710517), the Venture and Innovation Support Program for Chongqing Overseas Returnees (cx2022084) and the Fundamental Research Funds for the Central Universities (2023CDJXY-039).

References

[1] D. K. Naik and R. J. Mammone, "Meta-neural networks that learn by learning," in *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, vol. 1. IEEE, 1992, pp. 437–442.

[2] J. Liang, Y. Zhang, K. Chen, B. Qu, K. Yu, C. Yue, and P. N. Suganthan, "An evolutionary multiobjective method based on dominance and decomposition for feature selection in classification," *Science China Information Sciences*, vol. 67, no. 2, p. 120101, 2024.

[3] C. Finn, P. Abbeel, and S. Levine, Eds., *Model-agnostic meta-learning for fast adaptation of deep networks*. PMLR, 2017.

[4] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," in *Conference on robot learning*. PMLR, 2017, pp. 357–368.

[5] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8420–8429.

[6] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4367–4375.

[7] P. Yang, L. Zhang, H. Liu, and G. Li, "Reducing idleness in financial cloud services via multi-objective evolutionary reinforcement learning based load balancer," *Science China Information Sciences*, vol. 67, no. 2, pp. 1–21, 2024.

[8] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," *Advances in neural information processing systems*, vol. 29, 2016.

[9] 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.

[10] H. Yao, Y. Wei, J. Huang, and Z. Li, Eds., *Hierarchically structured meta-learning*. PMLR, 2019.

[11] H. Hao, X. Zhang, and A. Zhou, "Enhancing saeas with unevaluated solutions: a case study of relation model for expensive optimization," *Science China Information Sciences*, vol. 67, no. 2, pp. 1–18, 2024.

[12] C. Finn, K. Xu, and S. Levine, "Probabilistic model-agnostic meta-learning," *Advances in neural information processing systems*, vol. 31, 2018.

[13] C. Nguyen, T.-T. Do, and G. Carneiro, "Uncertainty in model-agnostic meta-learning using variational inference," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 3090–3100.

[14] S. Ravi and A. Beatson, "Amortized bayesian meta-learning," in *International Conference on Learning Representations*, 2019.

[15] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn, "Bayesian model-agnostic meta-learning," *Advances in neural information processing systems*, vol. 31, 2018.

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

[17] Z. Wang, J. Grigsby, A. Sekhon, and Y. Qi, Eds., *ST-MAML: A stochastic-task based method for task-heterogeneous meta-learning*. PMLR, 2022.

[18] P. Zhou, Y. Zou, X.-T. Yuan, J. Feng, C. Xiong, and S. Hoi, "Task similarity aware meta learning: Theory-inspired improvement on maml," in *Uncertainty in Artificial Intelligence*. PMLR, 2021, pp. 23–33.

[19] M. A. Jamal and G.-J. Qi, "Task agnostic meta-learning for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 719–11 727.

[20] H. Yao, X. Wu, Z. Tao, Y. Li, B. Ding, R. Li, and Z. Li, "Automated relational meta-learning," *arXiv preprint arXiv:2001.00745*, 2020.

[21] J.-Y. Li, Z.-H. Zhan, K. C. Tan, and J. Zhang, "A meta-knowledge transfer-based differential evolution for multitask optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 719–734, 2021.

[22] Y. Jiang, Z.-H. Zhan, K. C. Tan, and J. Zhang, "Block-level knowledge transfer for evolutionary multitask optimization," *IEEE Transactions on Cybernetics*, 2023.

[23] S.-H. Wu, Z.-H. Zhan, K. C. Tan, and J. Zhang, "Transferable adaptive differential evolution for many-task optimization," *IEEE Transactions on Cybernetics*, 2023.

[24] Q. Liu and D. Wang, "Stein variational gradient descent: A general purpose bayesian inference algorithm," *Advances in neural information processing systems*, vol. 29, 2016.

[25] K. Go, M. Kim, and S. Yun, "Meta-learning amidst heterogeneity and ambiguity," *IEEE Access*, vol. 11, pp. 1578–1592, 2022.

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