META-KNOWLEDGE EXTRACTION: UNCERTAINTY-AWARE PROMPTED META-LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Conventional meta-learning typically involves adapting all meta-knowledge to specific tasks, which incurs high computational costs due to the adaption process. To address this limitation, we introduce a more efficient gradient-based meta-learning framework called Uncertainty-Aware Prompted Meta-Learning (UAPML). Instead of adapting the entire meta-knowledge, we introduce a meta-knowledge extraction paradigm inspired by the success of large language models. In this paradigm, we freeze the model backbone and employ task-specific prompts to extract meta-knowledge for few-shot tasks. To construct the task-specific prompts, a learnable Bayesian meta-prompt is employed to provide an ideal initialization. Through theoretical analysis, we demonstrate that the posterior uncertainty of the Bayesian meta-prompt aligns with that of the task-specific prompt, which can be used to modulate the construction of task-specific prompts. Accordingly, we propose two ways, i.e., the soft and hard way, to automatically construct task-specific prompts from the meta-prompt when dealing with new tasks. Experimental results demonstrate the efficiency of the meta-knowledge extraction paradigm and highlight the significantly reduced computational cost achieved by our UAPML framework without the degradation of performance.

1 INTRODUCTION

Meta-learning (Hospedales et al., 2021; Huisman et al., 2021) has gained significant attention for its practicality in scenarios when only a limited number of labeled data are available. This techique inductively transfers meta-knowledge, such as parameter initialization, among different tasks, thereby improving data efficiency in low-resource scenarios. Most successful methods (Finn et al., 2017; Lee et al., 2019; Rusu et al., 2019), particularly optimization-based methods, mainly focus on adapting the meta-knowledge to task-specific knowledge, leading to different model variants for each specific task. Although such a knowledge adaption paradigm has yielded promising results, it comes at a high computational cost during the adaption process (Nichol et al., 2018).

The recent surge in popularity of large-scale pre-trained models such as ChatGPT and GPT-4 show a more efficient way to deal with tasks through knowledge extraction (Brown et al., 2020; Liu et al., 2023a). In contrast to knowledge adaption, knowledge extraction employs prompts (e.g., continuous tensors) that are prepended to the input, to bridge the gap between the general knowledge and task-specific knowledge. This efficient paradigm freezes the model backbone, eliminating the need for separate copies of the model for each task and significantly reducing the computational cost required for adaption on various tasks (Lester et al., 2021).

Therefore, we seek to leverage the knowledge extraction to overcome the computational inefficiency of the meta-knowledge adaption process (see Fig. 1). Applying knowledge extraction in the context of meta-learning requires two points (Liu et al., 2023b): (1) the compatibility of captured meta-knowledge across different tasks during the meta-training phase; (2) an appropriate prompt that effectively captures task-specific differences to enable task-specific knowledge extraction.

In response, we propose a novel **Uncertainty-Aware Prompted Meta-learning** (abbreviated as UAPML), which employs prompts for bridging the gap between general meta-knowledge and task-specific knowledge. In our UAPML framework, we share the entire model backbone (i.e., the feature extractor) among tasks to guarantee the compatibility of captured meta-knowledge, and design the task-specific prompt to help capture task-specific information. Sharing the backbone is supported by

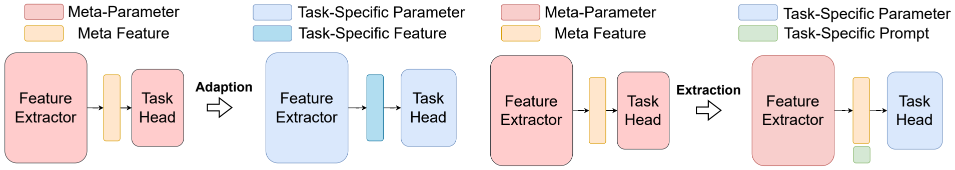


Figure 1: The comparison between meta-knowledge adaption and meta-knowledge extraction.

the previous findings that highlight the potential of shared features in the success of meta-learning (Raghu et al., 2019). It is also recognized that a fixed feature representation may not adequately cater to the diverse requirements of different tasks (Oh et al., 2020; Arnold & Sha, 2021), which confirms the need for the task-specific prompt. These task-specific prompts are continuous tensors that can be learned from available data, providing task-specific extraction in addition to the shared feature. However, learning distinct prompts from a large number of few-shot tasks is impractical and inefficient, and over-fitting easily occurs under the few-shot scenario. This pushes us to propose a Bayesian meta-prompt that provides an ideal initialization for task-specific prompts. Such a Bayesian treatment for meta-prompt also offers a crucial advantage of an uncertainty measure to reflect the relationship among tasks. Specifically, the posterior uncertainty of each dimension in the meta-prompt aligns with the uncertainty of task-specific prompts. With theoretical analysis, we show that highly shared dimensions among tasks exhibit low variance, indicating low uncertainty, while dimensions that are diverse and specific to certain tasks display higher variance, indicating higher uncertainty. Building upon this analysis, we introduce two automatic ways to derive the task-specific prompts from the meta-prompt: soft and hard modulation. The soft modulation adjusts the learning rate for each dimension when handling specific tasks, while the hard modulation freezes parts of dimensions for sharing and selectively tunes the remaining dimensions for specific tasks.

Our contributions can be summarized as follows. (1) We propose the Uncertainty-Aware Prompt Meta-learning (UAPML) framework, which improves the efficiency of gradient-based meta-learning by focusing on meta-knowledge extraction for few-shot tasks instead of computationally expensive meta-knowledge adaption (2) We provide a theoretical understanding of the alignment between Bayesian meta-prompt and task-specific prompt, enabling the construction of task-specific prompts that balance shared and task-specific information, enhancing flexibility in handling few-shot tasks (3) Based on our theoretical findings, we introduce two innovative methods for designing task-specific prompts using the Bayesian treatment of the meta-prompt. The soft and hard modulation techniques automatically generate task-specific prompts while considering the shared and task-specific information among the few-shot tasks. (4) We conduct extensive experiments in various settings, demonstrating that UAPML achieve a comparable or even prior performance over several state-of-the-art meta-learning methods when significantly reducing the computation consumption.

2 Related Work

Meta-Learning Meta-learning encompasses three main categories: optimization-based (Finn et al., 2017), model-based (Ha et al., 2016) and metric-based (Snell et al., 2017). We focus on the well-performed optimization-based meta-learning. These methods employ a meta-knowledge adaptation paradigm where all meta-knowledge is quickly adapted to specific tasks through gradient descent. However, this paradigm can be computationally expensive as all meta-knowledge needs to be adapted for each task. Efforts have been made to mitigate this burden. One (Nichol et al., 2018) uses first-order approximation, while another (Li et al., 2017) acquires an appropriate learning rate for one-step update adaptation. Recent work Raghu et al. (2019) has highlighted the significance of feature reuse in meta-learning success. By freezing most parameters except the prediction head, computational consumption can be significantly reduced. Another study (Von Oswald et al., 2021) follows a similar idea but incorporates sparse adaptation with additional masking parameters, enabling specific task adaptation of parameter subsets. While frozen representations have been effective in some cases, recent research by Oh et al. (2020) suggests that they may not adequately handle the complexity of real-world scenarios, highlighting the importance of embedding adaptation (Arnold & Sha, 2021). This trade-off between effectiveness and efficiency necessitates a solution. In our work, we address this challenge by incorporating prompts into the meta-learning framework. This approach provides an efficient means to capture task-specific information, bridging the gap between reused embeddings and the requirements of individual tasks.

Some other efforts also focus on capturing task-specific information, but their aims do not lie in efficiency. These efforts can be divided into two categories: mixture meta-knowledge (Yao et al., 2019; Jerfel et al., 2019; Yao et al., 2020; Zhang et al., 2021b; Wu et al., 2023) and conditional meta-learning (Rusu et al., 2019; Wang et al., 2020; Denevi et al., 2020). The mixture meta-knowledge approach associates clusters of similar tasks with different components, resulting in increased consumption proportional to the number of clusters. On the other hand, conditional meta-learning methods maintain a single meta-knowledge but need to design the additional architecture to customize the transfer of the meta-knowledge according to target tasks. For example, Rusu et al. (2019) proposes to employ an encoder-decoder architecture to construct task-specific classification heads, while Oreshkin et al. (2018) employs task-dependent scaled metric to customize the task-specific metric space. More recently, Baik et al. (2020) and Liu et al. (2020) employed a generative network to provide customized hyperparameters, such as the learning rate and the regularization weight, conditioned on the training state of the task. Since their aims do not lie in efficiency, their high consumption might be a concern. Instead, our proposed UAPML takes into account both effectiveness and efficiency, allowing a comparable or even better performance with less consumption.

Bayesian meta-learning is another relevant area. Previous efforts (Yoon et al., 2018; Gordon et al., 2019; Ravi & Beatson, 2019) have demonstrated the effectiveness of Bayesian treatment. Similarly, we employ Bayesian treatment on our meta-prompt but further conduct a comprehensive analysis of the benefits of the introduced uncertainty measure. Building upon the analysis, we propose two ways to construct the task-specific prompts, benefiting from the additional uncertainty measure.

Prompt Tuning Knowledge extraction (i.e., knowledge probing) (Petroni et al., 2019; Davison et al., 2019) is an important work in the language models. It involves using language triggers to induce the model to generate relational facts. Prompts, one important tool in knowledge extraction, is introduced in GPT-3 (Brown et al., 2020) and showed that can be designed or learned to capture the task-related information (Liu et al., 2021a). Schick & Schütze (2021) introduce pattern-exploiting training, demonstrating that providing task descriptions to pre-trained models can achieve comparable performance to standard supervised fine-tuning. Beyond discrete vocabulary spaces, many works try to learn prompt vectors in continuous spaces (Liu et al., 2021b; Li & Liang, 2021; Zhang et al., 2021a). These techniques, collectively referred to as prompt tuning (Lester et al., 2021; Gu et al., 2022), aim to optimize prompt representations for specific tasks, such as the semantic prompt captured from the class name Chen et al. (2023). In this paper, we are inspired by knowledge extraction and try to propose a more efficient meta-learning method, but not requiring extra context.

Efforts to integrate meta-learning with prompt tuning have been made in NLP. Hou et al. (2022) employ meta-learning to learn a better initialization while Jiang et al. (2023) further propose to construct a prompt pool. Different from the existing prompt tuning in NLP, we consider the meta-prompt as a stochastic high-dimensional variable, serving as an initialization for task-specific prompts. Additionally, we theoretically demonstrate that the posterior uncertainty of the meta-prompt can effectively guide the construction of task-specific prompts, which is a more efficient approach compared to prompt pools that necessitate linearly increasing resources.

3 Preliminary

Model-Agnostic Meta-learning Model-agnostic meta-learning (MAML) (Finn et al., 2017) is one of the most popular meta-learning methods. It employs a bi-level optimization framework to enable fast adaption of models across various few-shot tasks. Specifically, MAML aims to learn a decent initialization of model parameters, i.e., meta-knowledge, through the outer loop, while fine-tuning the model parameters to specific tasks through the inner adaption. This paradigm of adjusting meta-knowledge for different tasks can be considered as a knowledge adaption paradigm. Mathematically, let $\boldsymbol{\theta}$ be the initialization of model parameters, the inner adaption process of MAML yields a task-specific parameter $\boldsymbol{\theta}'$ for a few-shot task τ through one or more gradient descent updates, where one gradient descent update can be denoted as: $\boldsymbol{\theta}' = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_\tau(\boldsymbol{\theta})$, where α is the learning rate for the inner adaption and $\mathcal{L}_\tau$ is the task-specific loss function. The optimization objective of MAML is defined over a number of tasks: $\mathcal{L}(\boldsymbol{\theta}) = \int_{\tau \in p(\tau)} \mathcal{L}_\tau(\boldsymbol{\theta}') = \int_{\tau \in p(\tau)} \mathcal{L}_\tau(\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_\tau(\boldsymbol{\theta}))$. The objective is optimized via the outer loop: $\boldsymbol{\theta} = \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$, where β is the learning rate.

Prompt Tuning Prompt tuning (Lester et al., 2021; Gu et al., 2022) has emerged as a promising method for knowledge extraction, which utilizes pre-trained language models for various down-

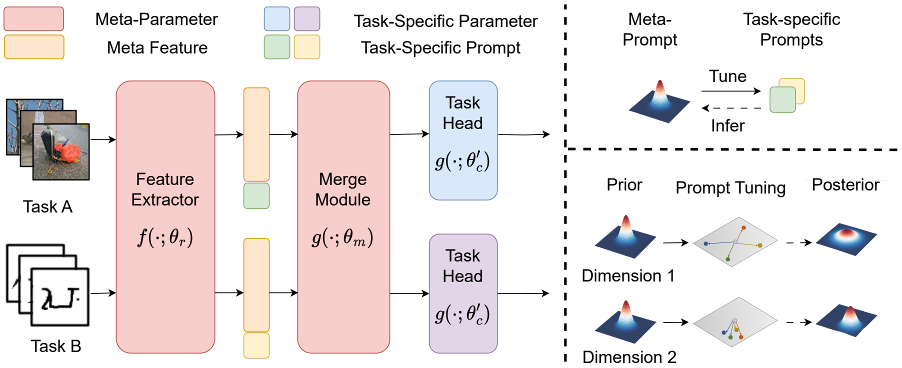


Figure 2: The architecture diagram of UAPML (**left**) and an intuitive illustration of how the uncertainty of the posterior meta-prompt is used to modulate the tuning of task-specific prompts (**right**).

stream tasks in the field of natural language processing. Instead of fine-tuning pre-trained language models on specific tasks, prompt tuning methods freeze the model backbone and incorporate task-specific prompts as additional inputs to effectively handle different tasks. The prompt is typically designed as a sequence of discrete tokens or implemented as a trainable continuous vector $\boldsymbol{s}$, which is then concatenated to the input of a pre-trained model to provide task-specific guidance. Specifically, suppose a pre-trained model $\mathcal{M}$ conducts a mapping $\pi : \mathcal{X} \rightarrow \mathcal{H}$, where $\mathcal{X}$ is the input space and $\mathcal{H}$ is the representation space. Normally, the pre-trained model outputs representations by: $\boldsymbol{h} = \pi(\boldsymbol{X})$, while the prompted model outputs representations by $\tilde{\boldsymbol{h}} = \pi(\boldsymbol{s}||\boldsymbol{X})$, where $||$ denotes the concatenation of inputs and prompts. The advantage of prompt tuning is that we do not need to update the parameters of the pre-trained model $\mathcal{M}$ but just learn different prompts when dealing with various downstream tasks.

4 Method

In what follows, we introduce the details of the proposed Uncertainty-Aware Prompted Meta-Learning (UAPML), which is shown in Fig. 2.

4.1 Prompted Meta-learning

Let the model parameters $\boldsymbol{\theta}$ consist of two parts, the model backbone $\boldsymbol{\theta}_r$ (e.g., the feature extractor) and the task classification head $\boldsymbol{\theta}_c$. Following the existing work (Raghu et al., 2019), our UAPML freezes the model backbone $\boldsymbol{\theta}_r$ and only adapts the classification head $\boldsymbol{\theta}_c$ to the specific tasks as $\boldsymbol{\theta}'_c$. This approach is based on the assumption, consistent with (Raghu et al., 2019), that the features extracted from the model backbone possess general meta-knowledge across various downstream few-shot tasks and should be reused. To facilitate the feature reuse considering the task-specific information, we employ a task-specific prompt to guide how to reuse the feature (i.e., the meta-knowledge extraction process) for a specific task.

4.1.1 Meta-Prompt

Since meta-learning models are designed to tackle a multitude of few-shot tasks, it would be inefficient or impractical to learn distinct prompts for each task. Additionally, directly learning such prompts on limited data can lead to overfitting issues. Therefore, following the bi-level optimization paradigm of meta-learning, we introduce a learnable meta-prompt $\boldsymbol{s} \in \mathbb{R}^n$, which serves as a decent initialization for task-specific prompts. To better adapt to real-world scenarios where tasks can significantly differ from each other, we employ a Bayesian treatment for the meta-prompt. We model the meta-prompt as a random variable instead of a deterministic vector. This Bayesian treatment allows us to capture the uncertainty associated with the meta-prompt and provides a flexible initialization for task-specific prompts. Such uncertainty measure introduced by Bayesian treatment can avoid conflict among tasks and help the model to better handle the variations in few-shot tasks.

Specifically, given a globally-shared meta-prompt distribution $p(\boldsymbol{s})$, we sample a meta-prompt from the distribution to provide initialization for task-specific prompts. For a specific downstream task, the meta-prompt is tuned with one or more steps of gradient descent updates:

$$\boldsymbol{s}' = \boldsymbol{s} - \alpha \nabla_{\boldsymbol{s}} \mathcal{L}_{\tau}(\boldsymbol{s}, \boldsymbol{\theta}), \;\; \boldsymbol{s} \sim p(\boldsymbol{s}). \tag{1}$$

To incorporate task-specific guidance, the task-specific prompts $\boldsymbol{s}'$ are concatenated with the features from the backbone model $\boldsymbol{\theta_r}$. This concatenation combines the knowledge encoded in the prompts and the features, enabling the model to leverage both sources of information for each specific task. Moreover, to fuse the information from the prompt and the backbone model, we introduce a merge module. The merge module is parameterized by a neural network with parameters $\boldsymbol{\theta}_m$. It takes both the task-specific prompts and the features as inputs. Such a merge module is designed to capture the complementary information between the prompts and the features, allowing them to effectively work together to improve the model's performance on the specific task:

$$\hat{\boldsymbol{y}} = g(f(\boldsymbol{x}; \boldsymbol{\theta}_r) || \boldsymbol{s}'; \boldsymbol{\theta}_m, \boldsymbol{\theta}'_c), \tag{2}$$

where f is the feature extractor and g is the merged layer with a task-specific classification head $\boldsymbol{\theta}'_c$. And the task-specific classification head is attained following:

$$\boldsymbol{\theta}'_c = \boldsymbol{\theta}_c - \alpha \nabla_{\boldsymbol{\theta}_c} \mathcal{L}_{\tau}(\boldsymbol{s}, \boldsymbol{\theta}), \;\; \boldsymbol{s} \sim p(\boldsymbol{s}). \tag{3}$$

Note that UAPML freezes the model backbone $\boldsymbol{\theta}_r$, occupying most of the parameters, as well as a small merge layers $\boldsymbol{\theta}_m$. **Only the meta-prompt $\boldsymbol{s}$ and the classification head $\boldsymbol{\theta}_c$ are updated for a specific task.** In this way, the computational consumption is significantly reduced. The analysis of the time complexity is shown in Appendix A.3. By combining the task-specific prompts, the features from the backbone model, and the merge module, the model can leverage the prompt-guided knowledge from the rich representations captured by the backbone model to make accurate predictions on the task at hand.

4.1.2 Inference Algorithm

With a set of meta-learning tasks $\mathcal{D}$, the Bayesian treatment on meta-prompt brings the issue of the intractability of the posterior $p(\boldsymbol{s}|\mathcal{D})$. To address this issue, we employ variational inference to approximately infer the posterior. Specifically, we approximate the posterior $p(\boldsymbol{s}|\mathcal{D})$ with a variational distribution $q(\boldsymbol{s}; \boldsymbol{\phi})$. The evidence lower bound is defined as:

$$\log p(\mathcal{D}; \boldsymbol{\theta}) \geq \mathbb{E}_{q(\boldsymbol{s};\boldsymbol{\phi})} \left[\log p(\mathcal{D}|\boldsymbol{s}; \boldsymbol{\theta})\right] - \mathbf{KL}\left[q(\boldsymbol{s}; \boldsymbol{\phi}) || p(\boldsymbol{s})\right], \tag{4}$$

where $KL(\cdot||\cdot)$ is the Kullback–Leibler divergence (KL-divergence) and $\log p(\mathcal{D}|\boldsymbol{s}; \boldsymbol{\theta})$ is the log-likelihood of data conditioned on the meta-prompt:

$$p(\mathcal{D}|\boldsymbol{s}; \boldsymbol{\theta}) = \int_{\tau \in p(\tau)} -\mathcal{L}_{\tau}(\boldsymbol{s}', \boldsymbol{\theta}), \tag{5}$$

where $\boldsymbol{s}'$ is the task-specific prompt after a few steps of gradient descent from the meta-prompt $\boldsymbol{s} \sim q(\boldsymbol{s}; \boldsymbol{\phi})$. Note that we assume $q(\boldsymbol{s}; \boldsymbol{\phi})$ to be a Gaussian with diagonal covariance, which is parameterized by $\boldsymbol{\phi} = \boldsymbol{\mu}, \boldsymbol{\rho}$. The standard deviation is parameterized as $\boldsymbol{\sigma} = \log(1 + \exp(\boldsymbol{\rho}))$ to avoid the invalid value (i.e., the negative value). The detailed derivation is shown in Appendix A.1.

4.2 Theoretical Analysis

The Bayesian meta-prompt serves as a valuable initialization for efficiently constructing task-specific prompts. However, the SGD-based construction approach treats each dimension equally, which may not be optimal (Kingma & Ba, 2014). This is because different dimensions of the meta-prompt can contain different amounts of shared information across tasks and require different levels of tuning to capture task-specific information. In this subsection, we present a theoretical analysis that demonstrates the alignment between the uncertainty of the meta-prompt and task-specific prompt when optimizing the objective Eq. 4. This finding motivates us to leverage the uncertainty of the meta-prompt to modulate the construction of task-specific prompts. In what follows, we provide the mathematical definition and the proof of this finding. More details are in Appendix A.2.

Definition 1 *Let $H(\cdot)$ be the entropy, which measures the uncertainty of the random variable. $H(\boldsymbol{s})$ and $H(\boldsymbol{s'}|\boldsymbol{s}, \mathcal{D})$ describe the uncertainty of meta-prompt and the task-specific prompts respectively.*

Proposition 1 *The uncertainty of the posterior of meta-prompt $H(\boldsymbol{s}_d)$ is in alignment with the uncertainty of task-specific prompts among tasks $H(\boldsymbol{s}'_d|\boldsymbol{s}_d, \mathcal{D})$ on each dimension d.*

To better present our analysis, we use another view of the optimization objective but equal to Eq. 4 following (Esmaeili et al., 2019). The objective of our proposed UAPML is to minimize the KL-divergence of each variable between the approximate posterior and the prior, which equals maximizing the negative KL-divergence (the lower bound $\mathcal{L}_{LB}$) and then can be derived as follows:

$$\begin{aligned}\mathcal{L}_{LB}(\boldsymbol{\theta},\boldsymbol{\phi}) = -\,\mathbf{KL}\left[q(\mathcal{D},\boldsymbol{s}',\boldsymbol{s};\boldsymbol{\theta},\boldsymbol{\phi})\|p(\mathcal{D},\boldsymbol{s}',\boldsymbol{s})\right] &= \mathbb{E}_{q(\mathcal{D},\boldsymbol{s}',\boldsymbol{s};\boldsymbol{\theta},\boldsymbol{\phi})}\left[\log\frac{p(\mathcal{D}|\boldsymbol{s}',\boldsymbol{s})}{p(\mathcal{D})} - \log\frac{q(\boldsymbol{s}',\boldsymbol{s}|\mathcal{D};\boldsymbol{\theta},\boldsymbol{\phi})}{q(\boldsymbol{s}',\boldsymbol{s};\boldsymbol{\theta},\boldsymbol{\phi})}\right] \\ &- \mathbf{KL}\left[q(\mathcal{D})\|p(\mathcal{D})\right] - \mathbf{KL}\left[q(\boldsymbol{s}'|\boldsymbol{s};\boldsymbol{\theta})\|p(\boldsymbol{s}'|\boldsymbol{s})\right] - \mathbf{KL}\left[q(\boldsymbol{s};\boldsymbol{\phi})\|p(\boldsymbol{s})\right], \qquad (6)\end{aligned}$$

where $q(D)$ and $p(D)$ are the accessible and the inaccessible data distribution in the real world, respectively (Esmaeili et al., 2019). Note that the expectation of the term $\log\frac{q(\boldsymbol{s}',\boldsymbol{s}|\mathcal{D};\boldsymbol{\theta},\boldsymbol{\phi})}{q(\boldsymbol{s}',\boldsymbol{s};\boldsymbol{\theta},\boldsymbol{\phi})}$ in Eq. 6 equals to the mutual information $I(\boldsymbol{s}',\boldsymbol{s};\mathcal{D})$ between prompts conditioned on the dataset $\mathcal{D}$ and not, and is derived as:

$$I(\boldsymbol{s}',\boldsymbol{s};\mathcal{D}) = H(q(\boldsymbol{s}'|\boldsymbol{s},\mathcal{D};\boldsymbol{\theta})) - H(q(\boldsymbol{s}'|\boldsymbol{s};\boldsymbol{\theta})) \approx H(q(\boldsymbol{s}'|\boldsymbol{s},\mathcal{D};\boldsymbol{\theta})) - H(q(\boldsymbol{s};\boldsymbol{\phi})) \qquad (7)$$

Note that the term $q(\boldsymbol{s}'|\boldsymbol{s},\mathcal{D};\boldsymbol{\theta})$ denotes the task-specific prompts given the meta-prompt among a set of meta-train tasks, as defined in Eq. 1. Since the term $q(\boldsymbol{s}'|\boldsymbol{s};\boldsymbol{\theta})$ can be seen as the ones without tuning, it equals to the meta-prompt that is utilized as the initialization, i.e., $q(\boldsymbol{s}'|\boldsymbol{s};\boldsymbol{\theta}) \approx q(\boldsymbol{s};\boldsymbol{\phi})$. This approximation supports the derivation of Eq. 7. Furthermore, the derived result can be decomposed into each dimension based on our independence dimension assumption:

$$H(q(\boldsymbol{s}'|\boldsymbol{s},\mathcal{D};\boldsymbol{\theta})) - H(q(\boldsymbol{s};\boldsymbol{\theta})) = \sum\nolimits_d H(q(\boldsymbol{s}'_d|\boldsymbol{s}_d,\mathcal{D};\boldsymbol{\theta})) - H(q(\boldsymbol{s}_d;\boldsymbol{\theta})) \qquad (8)$$

When a specific dimension of the task-specific prompts exhibits diversity among a set of tasks, the corresponding dimension of the meta-prompt will have a larger uncertainty, which is mainly reflected in a larger standard deviation in our Gaussian distribution assumption. This alignment of uncertainty provides a way to effectively modulate the tuning of task-specific prompts, to allow the dimension with larger uncertainty to capture more task information while allowing the dimensions with smaller uncertainty to maintain the common information.

4.3 Uncertainty-Aware Construction for prompts

Based on the aforementioned analysis, we propose two approaches, namely the soft and hard modulation, to effectively construct task-specific prompts from the meta-prompt. Note that these prompt construction methods are specifically designed for meta-test tasks, where the posterior distribution of the meta-prompt has already been inferred from a number of tasks.

Soft Modulation The soft modulation aims to encourage dimensions in the meta-prompt that contain more common information (i.e., dimensions with low uncertainty) to remain unchanged, thereby enabling information sharing across tasks. On the other hand, dimensions that share less common information are allowed to tune more, enabling the capture of task-specific information. Specifically, the soft construction regulates the learning rate on each dimension when tuning the prompts to specific tasks. We use $1/\boldsymbol{\sigma}_d$ to regulate the learning rate where $\boldsymbol{\sigma}_d$ is the standard deviation of d-th dimension of the meta-prompt:

$$h_{\text{soft}}(\alpha,\boldsymbol{\rho}) = \alpha/(1/\boldsymbol{\sigma}) = \alpha\log(1+\exp(\boldsymbol{\rho})). \qquad (9)$$

Such a method increases the learning rate of the dimensions with a large uncertainty to encourage to capture more task information while limiting the changes of the dimension with a low uncertainty to maintain the common information.

Hard Modulation The hard modulation approach follows a similar principle to the soft modulation but takes a more aggressive approach. It preserves dimensions in the meta-prompt with low uncertainty while allowing dimensions with higher uncertainty to freely adjust for task-specific information. It is similar to weight pruning (Han et al., 2015), a technique commonly used in network compression (Li et al., 2022), continual learning (Kang et al., 2022) and meta-learning (Von Oswald et al., 2021) to reduce computation and avoid over-parameterization. The previous weight tuning in meta-learning is achieved by freezing some weights in the inner adaption (Von Oswald et al., 2021) via learnable scores that weigh the importance of weights. This can lead to training instability due to the dynamic interaction between the scores and model parameters (Rosenbaum et al., 2019). Instead, we utilize the posterior inferred from meta-training, which is fixed when dealing with new tasks, to score the weights and make pruning decisions. Specifically, we employ the signal-to-noise ratio (SNR) (Blundell et al., 2015) , calculated as $\frac{|\boldsymbol{\mu}|}{\boldsymbol{\sigma}}$, to determine the amount of common information. The SNR takes into account both the standard deviation and mean value of the weights, where a larger mean indicates a more important role when merged with the representation. Consequently, weights with a large SNR will have a learning rate of zero to preserve the common information:

$$h_{\text{hard}}(\alpha, \boldsymbol{\mu}, \boldsymbol{\rho}; \gamma) = \alpha \cdot \mathbb{1}\{\frac{|\boldsymbol{\mu}|}{\boldsymbol{\sigma}} < \gamma\} = \alpha \cdot \mathbb{1}\{\frac{|\boldsymbol{\mu}|}{\log(1 + \exp(\boldsymbol{\rho}))} < \gamma\}, \tag{10}$$

where γ is the hyperparameter that controls the pruning rate and $\mathbb{1}$ is the indicator function. With the above soft and hard prompt construction methods, we can modulate the tuning of task-specific prompts to deal with new tasks based on the uncertainty of the posterior meta-prompt. The tuning for task-specific prompts in Eq. 1 can be reformulated as

$$\begin{cases} \boldsymbol{s}' = \boldsymbol{s} - h_{\text{soft}}(\alpha, \boldsymbol{\rho}) \nabla_{\boldsymbol{s}} \mathcal{L}_{\tau}(\boldsymbol{s}, \boldsymbol{\theta}), & \boldsymbol{s} \sim q(\boldsymbol{s}; \boldsymbol{\phi}), \\ \boldsymbol{s}' = \boldsymbol{s} - h_{\text{hard}}(\alpha, \boldsymbol{\mu}, \boldsymbol{\rho}; \gamma) \nabla_{\boldsymbol{s}} \mathcal{L}_{\tau}(\boldsymbol{s}, \boldsymbol{\theta}), & \boldsymbol{s} \sim q(\boldsymbol{s}; \boldsymbol{\phi}), \end{cases} \tag{11}$$

which are for the soft and hard construction for tasks-specific prompts, respectively.

5 Experiment

In what follows, we focus on three research problems: (1) What is the effectiveness and efficiency of the meta-knowledge extraction method (i.e., UAPML), especially compared to the meta-knowledge adaption methods? (2) How does the learnable Bayesian meta-prompt affect knowledge extraction? (3) Can our uncertainty-aware construction for task-specific prompts capture the task-specific information? To answer the above research questions, we compare the performance on the few-shot classification of our proposed UAPML and the following state-of-the-art baselines: (1) MAML (Finn et al., 2017): a classic meta-knowledge adaption method. (2) ANIL (Raghu et al., 2019): a more efficient method based on MAML but only adapts the classifier to tasks. (3) BOIL (Oh et al., 2020): a method instead adapts the backbone of the model during the inner adaption. (4) ProtoNet (Snell et al., 2017): a classic embedding-based meta-learning method. (5) Meta-SGD (Li et al., 2017): an efficient method that additionally learns the learning rate to adjust the one-step inner adaption. (6) Sparse-MAML (Von Oswald et al., 2021): a state-of-the-art method aims to learn which weight to change in the inner adaption. (7) LEO (Rusu et al., 2019): a conditional meta-learning method that generates a task-specific classification head. (8) Sharp-MAML (Abbas et al., 2022): a state-of-the-art variant of MAML that employs sharpness-aware minimization to achieve better performance.

In our experiment, the classification tasks consist of four sub-datasets: Aircraft (Maji et al., 2013), CIFAR-FS (Bertinetto et al., 2018), Mini-Imagenet (Vinyals et al., 2016) and miniQuickDraw (Ha & Eck, 2017). This mixture task distribution contains the classification of grey-scale images, fine-grained images and coarse-grained images. We evaluate the performance of models using the 5-way 1-shot and 5-way 5-shot settings. For meta-training, tasks are randomly sampled from the four datasets, with each task containing five randomly selected classes. After training, the performance is evaluated on four datasets respectively. Moreover, we also conduct experiments on the vanilla datasets, Mini-Imagenet and Tiered-Imagenet (Ren et al., 2018) respectively, to confirm the effectiveness and efficiency of UAPML. More details about the experiment are shown in Appendix A.4.

5.1 Overall Performance

RQ1: The Overall Comparision As shown in in Tab. 1 and Tab. 2, our proposed UAPML, including UAPML$_{soft}$ and UAPML$_{hard}$, exhibits improved performance when trained with both Conv4 and ResNet12 architecture under both 5-shot and 1-shot settings. Specifically, compared to the meta-knowledge adaption methods (i.e., MAML, ANIL and BOIL), UAPML achieve a noticeable improvement of 3.5% and 5.5% under 5-shot and 1-shot setting when using the Conv4 architecture. Likewise, when employing a larger ResNet12 architecture, UAPML achieves an improvement of 1.6% and 3.4% under the 5-shot and 1-shot settings, respectively. Moreover, our proposed UAPML save a 58% and 29% when training the Conv4 architecture under 5-shot and 1-shot settings respectively, and such a time-saving increase to 76% and 55% when training the ResNet12 architecture. This confirms the effectiveness and efficiency of meta-knowledge extraction paradigm.

In terms of efficiency, while the embedding-based method ProtoNet requires the least computational resources, it demonstrates a substantial performance gap when compared to UAPML, especially under the 1-shot setting. Comparatively, our proposed UAPML requires similar computational resources as other efficient meta-learning methods such as Meta-SGD and Sparse-MAML,

Table 1: The 5-way **5-shot** comparison of average accuracy and the time consumption on 4-layer convolution network and 12-layer ResNet.

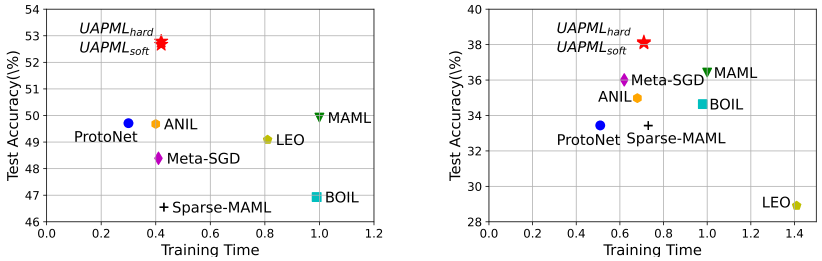
	Conv4			ResNet12		
	Average Accuracy	Training Time †	Adaption Time†	Average Accuracy	Training Time†	Adaption Time†
MAML	68.42$_{\pm 0.61}$	1.00	1.00	72.62$_{\pm 0.60}$	1.00	1.00
ANIL	68.40$_{\pm 0.59}$	0.40	0.58	72.33$_{\pm 0.61}$	0.22	0.34
BOIL	66.77$_{\pm 0.60}$	0.99	0.98	48.01$_{\pm 0.57}$	1.00	0.99
ProtoNet	64.74$_{\pm 0.60}$	0.30	0.14	72.78$_{\pm 0.54}$	0.15	0.10
Meta-SGD	58.83$_{\pm 0.61}$	0.41	0.34	53.88$_{\pm 0.59}$	0.31	0.30
Sparse-MAML	65.10$_{\pm 0.65}$	0.43	0.93	73.02$_{\pm 0.55}$	0.24	2.09
LEO	68.44$_{\pm 0.60}$	0.81	3.98	73.02$_{\pm 0.55}$	0.68	7.74
Sharp-MAML	66.61$_{\pm 0.58}$	3.28	2.27	73.49$_{\pm 0.58}$	4.16	4.91
UAPML$_{soft}$	$\mathbf{71.96}_{\pm \mathbf{0.59}}$	0.42	0.67	73.77$_{\pm 0.57}$	0.24	0.39
UAPML$_{hard}$	71.81$_{\pm 0.57}$	0.42	0.66	$\mathbf{74.25}_{\pm \mathbf{0.56}}$	0.24	0.39

† The time is normalized to MAML time.

Table 2: The 5-way **1-shot** comparison of average accuracy and the time consumption on 4-layer convolution network and 12-layer ResNet.

	Conv4			ResNet12		
	Average Accuracy	Training Time †	Adaption Time†	Average Accuracy	Training Time†	Adaption Time†
MAML	49.16$_{\pm 0.67}$	1.00	1.00	55.11$_{\pm 0.74}$	1.00	1.00
ANIL	51.09$_{\pm 0.69}$	0.68	0.60	54.44$_{\pm 0.74}$	0.41	0.37
BOIL	46.58$_{\pm 0.67}$	0.98	0.97	34.25$_{\pm 0.57}$	1.00	1.00
ProtoNet	43.34$_{\pm 0.61}$	0.51	0.12	54.82$_{\pm 0.69}$	0.35	0.10
Meta-SGD	42.47$_{\pm 0.62}$	0.62	0.35	41.71$_{\pm 0.64}$	0.47	0.31
Sparse-MAML	45.67$_{\pm 0.68}$	0.73	0.93	56.60$_{\pm 0.76}$	0.47	1.22
LEO	51.09$_{\pm 0.67}$	1.41	4.08	56.87$_{\pm 0.80}$	0.91	3.43
Sharp-MAML	49.42$_{\pm 0.68}$	2.83	2.18	56.60$_{\pm 0.76}$	3.86	2.94
UAPML$_{soft}$	$\mathbf{56.84}_{\pm \mathbf{0.73}}$	0.71	0.61	$\mathbf{58.52}_{\pm \mathbf{0.77}}$	0.45	0.38
UAPML$_{hard}$	56.36$_{\pm 0.73}$	0.71	0.61	57.56$_{\pm 0.77}$	0.45	0.38

† The time is normalized to MAML time.



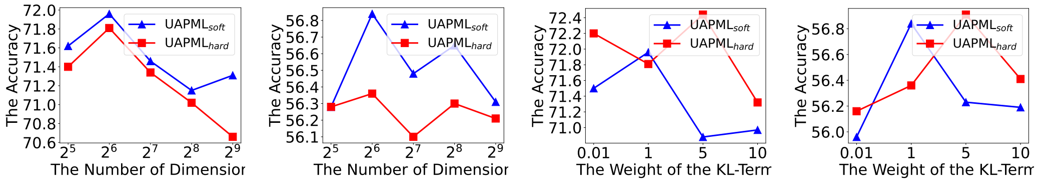
(a) 5-way 5-shot on Tiered-Imagenet

(b) 5-way 1-shot on Tiered-Imagenet

Figure 3: The 5-way 5-shot and 5-way 1-shot accuracy and training time of on Tiered-Imagenet.

but achieves superior performance. Furthermore, when compared to state-of-the-art methods such as LEO and Sharp-MAML, UAPML demonstrates significant improvements in performance across all settings, with the exception of the 5-shot evaluation on ResNet12 where UAPML still exhibits a marginal increase. This further confirms the efficiency and effectiveness of UAPML. More details about the test evaluation on each sub-dataset can be seen in Appendix A.5.

Furthermore, we provide results for all methods in Fig. 3, excluding Sharp-MAML due to its high computational requirements, on the vanilla datasets Tiered-Imagenet. The results on the Mini-Imagenet dataset follow a similar trend and are included in the Appendix A.5. The shown results illustrate that our UAPML still outperforms most of the baselines and even achieves a comparable performance compared to MAML which requires a much larger consumption. This confirms that UAPML would be the best choice when considering both effectiveness and efficiency.



(a) 5-way 5-shot (b) 5-way 1-shot (c) 5-way 5-shot (d) 5-way 1-shot

Figure 4: The average performance on four datasets with different dimensions (**left**) and with different weight of the KL-term (**right**) under 5-shot and 1-shot settings.

Table 3: The result of the ablation study.

	UAPML$_{soft}$	UAPML$_{hard}$	w/o Bayes	w/o C
5-shot	$71.96_{\pm 0.59}$	$71.81_{\pm 0.57}$	$70.53_{\pm 0.60}(\downarrow)$	$71.13_{\pm 0.58}(\downarrow)$
1-shot	$56.84_{\pm 0.73}$	$56.36_{\pm 0.73}$	$55.47_{\pm 0.74}(\downarrow)$	$55.82_{\pm 0.73}(\downarrow)$

5.2 The Affect of the Proposed Prompt

RQ2: The affect of Bayesian Meta-Prompt To further investigate the effectiveness of our proposed Bayesian meta-prompt, we conducted experiments with varying hyper-parameters to explore its design. We vary the number of the dimension of the meta-prompt from 2^5 to 2^9. The results, depicted in Fig. 4, indicate that under the 5-shot setting, a smaller dimension (e.g., from 2^6 to 2^5) led to a nearly 0.5% decrease in performance, while increasing the dimension resulted in a significant reduction of more than 1%. Similar patterns were observed under the 1-shot setting, where a smaller dimension negatively impacted performance, and a larger dimension did not yield performance improvements and could even have a detrimental effect. The weight assigned to the KL-term (see Eq. 4) in Bayesian inference played a crucial role, denoting the importance of the uncertainty measure. Following prior research (Higgins et al., 2017), we examined the impact of enlarging and reducing the weight of the KL-term on the performance of our approach (UAPML). As illustrated in Fig. 4, a smaller weight for the KL-term resulted in worse performance, except for UAPML$_{hard}$. On the other hand, a larger weight improved the performance of both UAPML$_{soft}$ and UAPML$_{hard}$, although excessively large weights (e.g., weight=10) had a negative effect. These findings confirm that the Bayesian treatment effectively captures task uncertainty to enhance performance, and incentivizing the capture of uncertainty in the meta-prompt is beneficial for hard construction. Additionally, we conducted an ablation study by removing the Bayesian treatment (i.e., w/o Bayes in Tab. 3), which led to a decrease in average performance ranging from nearly 1% to 1.5%. This further supports the effectiveness of the Bayesian treatment on the meta-prompt.

RQ3: The Construction of Task-Specific Prompts We further conduct experiments to explore our proposed uncertainty-aware constructions for task-specific prompts. To achieve this aim, we first evaluate the variant of our methods, which constructs task-specific prompts without any modulation (i.e., w/o C). The results shown in Tab. 3 illustrate that there is a reduction of nearly 1% on the way of construction without any uncertainty-aware modulation. It confirms that the uncertainty-aware construction for task-specific prompts helps to improve the performance on new tasks.

6 Conclusion

This paper introduces UAPML, an efficient meta-learning method that employs meta-knowledge extraction instead of conventional meta-knowledge adaptation. The method incorporates a task-specific prompt to facilitate efficient utilization of acquired meta-knowledge and utilizes a learnable Bayesian meta-prompt to enable the quick adaption of task-specific prompts on new tasks. Furthermore, by analyzing the posterior uncertainty of the meta-prompt, we propose soft and hard modulation techniques to construct task-specific prompts, striking a balance between capturing task-specific information and maintaining shared knowledge. Experimental results demonstrate that UAPML outperforms other meta-learning methods even with less consumption, highlighting the effectiveness and efficiency of the meta-knowledge extraction approach.

UAPML significantly contributes to the progress of meta-learning by offering an efficient approach to leveraging meta-knowledge and establishing a more effective and efficient framework for few-shot learning. This research opens up possibilities for further exploration and improvement in the field of meta-learning and paves the way for developing more powerful and flexible learning systems.

REFERENCES

Momin Abbas, Quan Xiao, Lisha Chen, Pin-Yu Chen, and Tianyi Chen. Sharp-maml: Sharpness-aware model-agnostic meta learning. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10–32. PMLR, 2022.

Sébastien MR Arnold and Fei Sha. Embedding adaptation is still needed for few-shot learning. *arXiv preprint arXiv:2104.07255*, 2021.

Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. Meta-learning with adaptive hyperparameters. *Advances in neural information processing systems*, 33:20755–20765, 2020.

Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Wentao Chen, Chenyang Si, Zhang Zhang, Liang Wang, Zilei Wang, and Tieniu Tan. Semantic prompt for few-shot image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23581–23591, 2023.

Joe Davison, Joshua Feldman, and Alexander M. Rush. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 1173–1178. Association for Computational Linguistics, 2019.

Giulia Denevi, Massimiliano Pontil, and Carlo Ciliberto. The advantage of conditional meta-learning for biased regularization and fine tuning. *Advances in Neural Information Processing Systems*, 33:964–974, 2020.

Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, Narayanaswamy Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem Meent. Structured disentangled representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2525–2534. PMLR, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E. Turner. Meta-learning probabilistic inference for prediction. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. Ppt: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8410–8423, 2022.

David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 5149–5169, 2021.

Yutai Hou, Hongyuan Dong, Xinghao Wang, Bohan Li, and Wanxiang Che. Metaprompting: Learning to learn better prompts. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pp. 3251–3262. International Committee on Computational Linguistics, 2022.

Mike Huisman, Jan N Van Rijn, and Aske Plaat. A survey of deep meta-learning. *Artificial Intelligence Review*, 54(6):4483–4541, 2021.

Ghassen Jerfel, Erin Grant, Tom Griffiths, and Katherine A Heller. Reconciling meta-learning and continual learning with online mixtures of tasks. *Advances in Neural Information Processing Systems*, 32, 2019.

Weisen Jiang, Yu Zhang, and James T. Kwok. Effective structured prompting by meta-learning and representative verbalizer. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15186–15199. PMLR, 2023.

Haeyong Kang, Rusty John Lloyd Mina, Sultan Rizky Hikmawan Madjid, Jaehong Yoon, Mark Hasegawa-Johnson, Sung Ju Hwang, and Chang D Yoo. Forget-free continual learning with winning subnetworks. In *International Conference on Machine Learning*, pp. 10734–10750. PMLR, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10657–10665, 2019.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *CoRR*, abs/2101.00190, 2021. URL `https://arxiv.org/abs/2101.00190`.

Yawei Li, Kamil Adamczewski, Wen Li, Shuhang Gu, Radu Timofte, and Luc Van Gool. Revisiting random channel pruning for neural network compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 191–201, 2022.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586, 2021a.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023a.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. GPT understands, too. *CoRR*, abs/2103.10385, 2021b.

Yaoyao Liu, Bernt Schiele, and Qianru Sun. An ensemble of epoch-wise empirical bayes for few-shot learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pp. 404–421. Springer, 2020.

Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. *Proceedings of the ACM Web Conference 2022*, 2023b.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Jaehoon Oh, Hyungjun Yoo, ChangHwan Kim, and Se-Young Yun. Boil: Towards representation change for few-shot learning. In *International Conference on Learning Representations*, 2020.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31, 2018.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 2463–2473. Association for Computational Linguistics, 2019.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations*, 2019.

Sachin Ravi and Alex Beatson. Amortized bayesian meta-learning. In *International Conference on Learning Representations*, 2019.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018.

Clemens Rosenbaum, Ignacio Cases, Matthew Riemer, and Tim Klinger. Routing networks and the challenges of modular and compositional computation. *arXiv preprint arXiv:1904.12774*, 2019.

Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Tetsuya Sakai. Two sample t-tests for ir evaluation: Student or welch? In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 1045–1048, 2016.

Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 255–269, 2021.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Johannes Von Oswald, Dominic Zhao, Seijin Kobayashi, Simon Schug, Massimo Caccia, Nicolas Zucchet, and João Sacramento. Learning where to learn: Gradient sparsity in meta and continual learning. *Advances in Neural Information Processing Systems*, 34:5250–5263, 2021.

Ruohan Wang, Yiannis Demiris, and Carlo Ciliberto. A structured prediction approach for conditional meta-learning. *Advances in Neural Information Processing Systems*, 2020.

Bin Wu, Jinyuan Fang, Xiangxiang Zeng, Shangsong Liang, and Qiang Zhang. Adaptive compositional continual meta-learning. In *International conference on machine learning*. PMLR, 2023.

Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In *International Conference on Machine Learning*, pp. 7045–7054. PMLR, 2019.

Huaxiu Yao, Yingbo Zhou, Mehrdad Mahdavi, Zhenhui Jessie Li, Richard Socher, and Caiming Xiong. Online structured meta-learning. *Advances in Neural Information Processing Systems*, 33:6779–6790, 2020.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. *Advances in neural information processing systems*, 31, 2018.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. Differentiable prompt makes pre-trained language models better few-shot learners. *CoRR*, abs/2108.13161, 2021a. URL `https://arxiv.org/abs/2108.13161`.

Qiang Zhang, Jinyuan Fang, Zaiqiao Meng, Shangsong Liang, and Emine Yilmaz. Variational continual bayesian meta-learning. *Advances in Neural Information Processing Systems*, 34:24556–24568, 2021b.

A Appendix

A.1 The ELBO of the Meta-Prompt

Given that we are employing the Bayesian approach to analyze the meta-prompt, it is essential to obtain its posterior distribution. However, obtaining the exact posterior distribution $p(\boldsymbol{s}|\mathcal{D})$ of the meta-prompt is infeasible. In order to address this issue, we resort to the variational inference method (Kingma & Welling, 2013) to derive an approximate distribution. Specifically, we employ a variational distribution $q(\boldsymbol{s};\boldsymbol{\phi})$ as an approximation to the true posterior. This variational distribution can be learned by maximizing the log-likelihood of all datasets $\log p(\mathcal{D};\boldsymbol{\theta})$, which can be achieved by maximizing the lower bound on the evidence (ELBO). The ELBO of the log-likelihood can be obtained through the application of Jensen's inequality:

$$\begin{aligned}
\log p(\mathcal{D};\boldsymbol{\theta}) &= \log \int p(\mathcal{D},\boldsymbol{s};\boldsymbol{\theta})d\boldsymbol{s} \\
&= \log \int p(\mathcal{D}|\boldsymbol{s};\boldsymbol{\theta})p(\boldsymbol{s})d\boldsymbol{s} \\
&= \log \int \frac{q(\boldsymbol{s};\boldsymbol{\phi})p(\mathcal{D}|\boldsymbol{s};\boldsymbol{\theta})p(\boldsymbol{s})}{q(\boldsymbol{s};\boldsymbol{\phi})}d\boldsymbol{s} \\
&\geq \mathbb{E}_{q(\boldsymbol{s};\boldsymbol{\phi})}\left[\log p(\mathcal{D}|\boldsymbol{s};\boldsymbol{\theta})\right] - \mathbf{KL}\left[q(\boldsymbol{s};\boldsymbol{\phi})||p(\boldsymbol{s})\right], \qquad (12)
\end{aligned}$$

where KL denotes the Kullback–Leibler divergence that is utilized as a measure of the dissimilarity between two probability distributions. In Equation 12, the first term represents the log-likelihood of all datasets conditioned on the variational distribution of the meta-prompt. The second term corresponds to the KL divergence between the variational posterior and the prior distribution of the meta-prompt.

Note that we assume $q(\boldsymbol{s};\boldsymbol{\phi})$ to be a Gaussian with diagonal covariance, which is parameterized by $\boldsymbol{\phi} = \boldsymbol{\mu},\boldsymbol{\rho}$. The standard deviation is parameterized as $\boldsymbol{\sigma} = \log(1+\exp(\boldsymbol{\rho}))$ to avoid the invalid value (i.e., the negative value). For the prior distribution of meta-prompts, we follow (Blundell et al., 2015) to use a mixture of Gaussian distribution, which consists of one standard Gaussian and another zero-centered Gaussian but with a larger variance to provide a heavier tail in the prior density. Since such a mixture prior focuses more on the tail part, it prevents the posterior from collapsing into some extreme points, leading to improved performance. We compute the ELBO in Eq. 4 using Monte Carlo sampling with the reparameterization trick (Kingma & Welling, 2013), where we sample from variational distributions to approximate the expectation of log-likelihood as well as the KL-term.

A.2 The Detailed Proof for the Proposition

In this subsection, we present a detailed derivation of the proof for Proposition 1. We adopt a different perspective to decouple the objective as follows:

$$\begin{aligned}
\mathcal{L}_{LB}(\boldsymbol{\theta},\boldsymbol{\phi}) =& -\mathbf{KL}\left[q(\mathcal{D},\boldsymbol{s}',\boldsymbol{s};\boldsymbol{\theta},\boldsymbol{\phi})\|p(\mathcal{D},\boldsymbol{s}',\boldsymbol{s})\right] \\
=&\mathbb{E}_{q(\mathcal{D},\boldsymbol{s}',\boldsymbol{s};\boldsymbol{\theta},\boldsymbol{\phi})}\left[\log\frac{p(\mathcal{D}|\boldsymbol{s}',\boldsymbol{s})p(\boldsymbol{s}'|\boldsymbol{s})p(\boldsymbol{s})q(\boldsymbol{s}',\boldsymbol{s};\boldsymbol{\theta},\boldsymbol{\phi})p(\mathcal{D})}{q(\boldsymbol{s}',\boldsymbol{s}|\mathcal{D};\boldsymbol{\theta},\boldsymbol{\phi})q(\mathcal{D})q(\boldsymbol{s}',\boldsymbol{s};\boldsymbol{\theta},\boldsymbol{\phi})p(\mathcal{D})}\right] \\
=&\mathbb{E}_{q(\mathcal{D},\boldsymbol{s}',\boldsymbol{s};\boldsymbol{\theta},\boldsymbol{\phi})}\left[\log\frac{p(\mathcal{D}|\boldsymbol{s}',\boldsymbol{s})}{p(\mathcal{D})} - \log\frac{q(\boldsymbol{s}',\boldsymbol{s}|\mathcal{D};\boldsymbol{\theta},\boldsymbol{\phi})}{q(\boldsymbol{s}',\boldsymbol{s};\boldsymbol{\theta},\boldsymbol{\phi})} - \log\frac{q(\mathcal{D})}{p(\mathcal{D})} - \log\frac{q(\boldsymbol{s}'|\boldsymbol{s};\boldsymbol{\theta})}{p(\boldsymbol{s}'|\boldsymbol{s})} - \log\frac{q(\boldsymbol{s};\boldsymbol{\phi})}{p(\boldsymbol{s})}\right] \\
=&\mathbb{E}_{q(\mathcal{D},\boldsymbol{s}',\boldsymbol{s};\boldsymbol{\theta},\boldsymbol{\phi})}\left[\log\frac{p(\mathcal{D}|\boldsymbol{s}',\boldsymbol{s})}{p(\mathcal{D})} - \log\frac{q(\boldsymbol{s}',\boldsymbol{s}|\mathcal{D};\boldsymbol{\theta},\boldsymbol{\phi})}{q(\boldsymbol{s}',\boldsymbol{s};\boldsymbol{\theta},\boldsymbol{\phi})}\right] \\
&-\mathbf{KL}\left[q(\mathcal{D})\|p(\mathcal{D})\right] - \mathbf{KL}\left[q(\boldsymbol{s}'|\boldsymbol{s};\boldsymbol{\theta})\|p(\boldsymbol{s}'|\boldsymbol{s})\right] - \mathbf{KL}\left[q(\boldsymbol{s};\boldsymbol{\phi})\|p(\boldsymbol{s})\right],
\end{aligned}$$

where the first term comprises of two components - the log-likelihood of all the datasets conditioned on the prompt and the ratio between the meta-prompt and task-specific prompt conditioned and unconditioned on all the datasets. The remaining three terms in the objective are KL-divergences between the variational distributions and the priors. In particular, we emphasize the latter part of the first term, which can be interpreted as the mutual information between the joint distribution of the meta-prompt and task-specific prompts and all the datasets. This mutual information can be further derived as:

$$\begin{aligned} I(\boldsymbol{s}', \boldsymbol{s}; \mathcal{D}) &= \mathbb{E}_{q(\boldsymbol{s}', \boldsymbol{s}, \mathcal{D})}\left[\log \frac{q(\boldsymbol{s}', \boldsymbol{s}|\mathcal{D}; \boldsymbol{\theta}, \boldsymbol{\phi})}{q(\boldsymbol{s}', \boldsymbol{s}; \boldsymbol{\theta}, \boldsymbol{\phi})}\right] \\ &= \mathbb{E}_{q(\boldsymbol{s}', \boldsymbol{s}, \mathcal{D})}\left[\log \frac{q(\boldsymbol{s}'|\boldsymbol{s}, \mathcal{D}; \boldsymbol{\theta}) q(\boldsymbol{s}; \boldsymbol{\phi})}{q(\boldsymbol{s}'|\boldsymbol{s}; \boldsymbol{\theta}) q(\boldsymbol{s}; \boldsymbol{\phi})}\right] \\ &= \mathbb{E}_{q(\boldsymbol{s}', \boldsymbol{s}, \mathcal{D})}\left[q(\boldsymbol{s}'|\boldsymbol{s}, \mathcal{D}; \boldsymbol{\theta})\right] - \mathbb{E}_{q(\boldsymbol{s}', \boldsymbol{s}, \mathcal{D})}\left[q(\boldsymbol{s}'|\boldsymbol{s}; \boldsymbol{\theta})\right] \\ &= H(q(\boldsymbol{s}'|\boldsymbol{s}, \mathcal{D}; \boldsymbol{\theta})) - H(q(\boldsymbol{s}'|\boldsymbol{s}; \boldsymbol{\theta})) \\ &\approx H(q(\boldsymbol{s}'|\boldsymbol{s}, \mathcal{D}; \boldsymbol{\theta})) - H(q(\boldsymbol{s}; \boldsymbol{\phi})), \end{aligned} \tag{13}$$

where the mutual information term can be separated into the difference between two information entropy. In the absence of datasets, the task-specific prompts retain their initializations that are drawn from the distribution of the meta-prompt, thereby supporting the above derivation. Consequently, one of the goals is to align the information entropy of the meta-prompt with that of the task-specific prompts.

Algorithm 1: The meta-train stage of UAPML.

```
Input: the meta-training datasets $p(\mathcal{D})$, the learning rate $\beta$ and $\alpha$ for the outer loop and inner loop,
    the number of inner steps $n$.
Output: the meta-parameters $\boldsymbol{\theta}_r$, $\boldsymbol{\theta}_m$ and $\boldsymbol{\theta}_c$, the posterior of the meta-prompt $q(\boldsymbol{s})$ parameterized
    by $\boldsymbol{\phi}$.
Initialize meta-parameters and the meta-prompt: $\boldsymbol{\theta}_r$, $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_c$ and $q(\boldsymbol{s}; \boldsymbol{\phi})$;
while not converage do
    Sample a tasks $\mathcal{D} \sim p(\mathcal{D})$;
    Sample a meta-prompt $\boldsymbol{s} \sim q(\boldsymbol{s}; \boldsymbol{\phi})$;
    Initialize the task-specific prompts with the sampled meta-prompt: $\boldsymbol{s'} \leftarrow \boldsymbol{s}$;
    for i = 1,2,..,n do
        Compute the Loss $\mathcal{L}(\boldsymbol{s}, \boldsymbol{\theta})$;
        Update the task-specific prompt $\boldsymbol{s}'$ according to Eq. 1;
        Update the task-specific classification $\boldsymbol{\theta}_c$ with the learning rate $\alpha$;
    end for
    Compute the ELBO according to Eq. 4;
    Update $\boldsymbol{\theta}_r$, $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_r$ and the variational parameters $\boldsymbol{\phi}$ of the meta-prompt with the learning
        rate $\beta$;
end while
```

Algorithm 2: The meta-test stage of UAPML

```
Input: the meta-parameters $\boldsymbol{\theta}_r$, $\boldsymbol{\theta}_m$ and $\boldsymbol{\theta}_c$, the posterior meta-prompt $q(\boldsymbol{s})$ parameterized by
    $\boldsymbol{\phi} = \{\boldsymbol{\mu}, \boldsymbol{\rho}\}$, the tuning rate $\alpha$, the number of inner steps $n$ and the task $\tau$.
Output: the task-specific prompts $\boldsymbol{s}'$ and the task-specific head $\boldsymbol{\theta}'_c$.
Sample an initialization for task-specific prompts from the posterior meta-prompt:
    $\boldsymbol{s}' \leftarrow \boldsymbol{s},\ \boldsymbol{s} \sim q(\boldsymbol{s}; \boldsymbol{\phi})$;
for $i = 1, 2, 3, ..., n$ do
    Compute the tuning rate $h(\alpha)$ according to Eq. 9 or Eq. 10;
    Tune the task-specific prompt $\boldsymbol{s}'$ and the task-specific head according to Eq. 11 and Eq. 3;
end for
```

A.3 The Analysis of the time complexity

The pseudo-code of the meta-train and meta-test stage of our proposed UAPML are shown in Alg.1 and Alg.2, respectively. One of the benefits of the meta-knowledge extraction paradigm is its efficiency. Hence, we analyze the time complexity of several popular meta-learning methods, compared to our proposed UAPML.

MAML (Finn et al., 2017), the most widely used meta-learning approach, employs a bi-level optimization scheme that involves an outer loop for updating meta-parameters and an inner loop for evaluating the current meta-parameters. Consequently, its time complexity is $\mathcal{O}(n^2)$ due to this bi-level optimization architecture. BOIL (Oh et al., 2020), like MAML, freezes only the classifier on the last layer and still updates most of the model parameters during the inner adaptation, resulting in a time complexity of $\mathcal{O}(n^2)$. ANIL (Raghu et al., 2019), on the other hand, updates only the last layer, i.e., the classifier, during the inner adaptation, leading to a reduced time complexity of $\mathcal{O}(n)$. Similarly, our proposed UAPML only tunes the last layer and a small-scale prompt, both of which are located on the last layers, resulting in significant computation reduction for gradients. Therefore, the time complexity of our proposed UAPML is similar to ANIL, i.e., $\mathcal{O}(n)$. The experimental results in the following section demonstrate the effectiveness of our proposed method in comparison to these state-of-the-art meta-learning approaches.

A.4 The Details of Experiment

A.4.1 The Datasets

In what follows, we present a detailed introduction of the datasets used in our experiments:

Aircraft: Aircraft (Maji et al., 2013) is a fine-grained dataset containing 100 aircraft model classes, each with 100 images. We randomly divide 64 classes for training, 16 classes for validation, and 20 classes for evaluation.

CIFAR-FS: CIFAR-FS (Bertinetto et al., 2018) is a well-known image classification dataset comprising 100 classes of objects, each with 600 images. We randomly select 100 images for each class and randomly divide 64 classes for training, 16 classes for validation, and 20 classes for evaluation.

miniImageNet: miniImageNet (Vinyals et al., 2016) is a popular dataset for meta-learning, which consists of 100 classes and 600 images for each class from the ImageNet dataset. We randomly select 100 images for each class and split 100 classes into 64 classes for training, 16 classes for validation, and 20 classes for evaluation.

miniQuickDraw: miniQuickDraw (Ha & Eck, 2017) is a grayscale image classification dataset containing 345 classes sampled from QuickDraw. Consistent with the other datasets, we randomly select 100 classes with each class containing 100 images. The selected classes are split into 64 classes for training, 16 classes for validation, and the remaining 20 classes for evaluation.

Tiered-Imagenet Tiered-Imagenet (Ren et al., 2018) is a larger subset of ILSVRC-12 (608 classes rather than 100 for miniImageNet). We randomly select 1300 images for each class, and split 608 classes into 351 classes for training, 97 classes for validation and the remaining 160 classes for evaluation.

A.4.2 The Baselines

To evaluate the effectiveness of our proposed UAPML and address the research problems, we compare our method with the following meta-learning approaches:

MAML (Finn et al., 2017): a classic method that learns a good initialization for model parameters and adapts fully to task-specific parameters, requiring a bi-level optimization architecture with an outer loop for updating the initialization and an inner loop for evaluating the initialization.

ANIL (Raghu et al., 2019): a method based on MAML that proposes to use feature reuse instead of quick adaptation to reduce computational overheads. It freezes the model backbone (i.e., feature extractor) and only updates the task head when dealing with few-shot tasks.

BOIL (Oh et al., 2020): a method that adapts the backbone of the model during the inner adaptation to handle different task distributions, so as to provide different features for tasks, which is in contrast to ANIL.

ProtoNet (Snell et al., 2017): a classic embedding-based meta-learning method that learns a metric space for few-shot tasks. Since it does not need the inner adaption, the computational consumption is much less compared to MAML.

Table 4: **5-way 5-shot** accuracy (%) with 95% confidence interval of Conv4 architecture on each task distribution and on average. The best and the second-best performances are bolded and underlined, respectively.

	Aircraft	CIFAR-FS	Mini-Imagenet	miniQuickDraw	Average
MAML	$69.00_{\pm 0.59}$	$67.71_{\pm 0.64}$	$59.27_{\pm 0.69}$	$\mathbf{77.68}_{\pm \mathbf{0.51}}$	$68.42_{\pm 0.61}$
ProtoNet	$64.06_{\pm 0.60}$	$63.21_{\pm 0.62}$	$57.03_{\pm 0.64}$	$74.64_{\pm 0.54}$	$64.74_{\pm 0.60}$
ANIL	$70.11_{\pm 0.59}$	$67.93_{\pm 0.60}$	$59.08_{\pm 0.65}$	$76.48_{\pm 0.53}$	$68.40_{\pm 0.59}$
BOIL	$68.20_{\pm 0.57}$	$65.98_{\pm 0.62}$	$58.82_{\pm 0.66}$	$74.10_{\pm 0.54}$	$66.77_{\pm 0.60}$
Meta-SGD	$55.94_{\pm 0.58}$	$57.05_{\pm 0.63}$	$52.95_{\pm 0.63}$	$69.39_{\pm 0.58}$	$58.83_{\pm 0.61}$
LEO	$70.10_{\pm 0.59}$	$68.35_{\pm 0.59}$	$58.52_{\pm 0.62}$	$76.76_{\pm 0.60}$	$68.44_{\pm 0.60}$
Sparse-MAML	$67.98_{\pm 0.61}$	$63.89_{\pm 0.68}$	$56.14_{\pm 0.70}$	$72.40_{\pm 0.62}$	$65.10_{\pm 0.65}$
Sharp-MAML	$68.06_{\pm 0.57}$	$66.43_{\pm 0.60}$	$56.34_{\pm 0.64}$	$75.62_{\pm 0.53}$	$66.61_{\pm 0.58}$
UAPML$_{soft}$	$\underline{76.32_{\pm 0.54}}$	$\mathbf{72.37}_{\pm \mathbf{0.60}}$	$\mathbf{61.83}_{\pm \mathbf{0.65}}$	$77.31_{\pm 0.55}$	$\mathbf{71.96}_{\pm \mathbf{0.59}}$
UAPML$_{hard}$	$\mathbf{76.53}_{\pm \mathbf{0.56}}$	$\underline{71.48_{\pm 0.56}}$	$\underline{61.59_{\pm 0.65}}$	$\underline{77.63_{\pm 0.52}}$	$\underline{71.81_{\pm 0.57}}$

Table 5: **5-way 1-shot** accuracy (%) with 95% confidence interval of Conv4 architecture on each task distribution and on average. The best and the second-best performances are bolded and underlined, respectively.

	Aircraft	CIFAR-FS	Mini-Imagenet	miniQuickDraw	Average
MAML	$47.80_{\pm 0.61}$	$48.18_{\pm 0.70}$	$42.17_{\pm 0.65}$	$58.49_{\pm 0.71}$	$49.16_{\pm 0.67}$
ProtoNet	$34.70_{\pm 0.51}$	$44.49_{\pm 0.65}$	$39.23_{\pm 0.60}$	$54.94_{\pm 0.69}$	$43.34_{\pm 0.61}$
ANIL	$53.71_{\pm 0.65}$	$48.88_{\pm 0.70}$	$44.03_{\pm 0.69}$	$57.72_{\pm 0.72}$	$51.09_{\pm 0.69}$
BOIL	$40.27_{\pm 0.58}$	$47.18_{\pm 0.69}$	$41.71_{\pm 0.68}$	$57.15_{\pm 0.74}$	$46.58_{\pm 0.67}$
Meta-SGD	$39.10_{\pm 0.56}$	$40.89_{\pm 0.65}$	$37.85_{\pm 0.61}$	$53.07_{\pm 0.65}$	$42.47_{\pm 0.62}$
LEO	$53.95_{\pm 0.65}$	$47.15_{\pm 0.65}$	$41.49_{\pm 0.63}$	$\mathbf{61.78}_{\pm \mathbf{0.75}}$	$51.09_{\pm 0.67}$
Sparse-MAML	$47.95_{\pm 0.66}$	$43.00_{\pm 0.72}$	$38.68_{\pm 0.66}$	$53.05_{\pm 0.70}$	$45.67_{\pm 0.68}$
Sharp-MAML	$51.30_{\pm 0.68}$	$46.69_{\pm 0.67}$	$41.62_{\pm 0.65}$	$58.08_{\pm 0.70}$	$49.42_{\pm 0.68}$
UAPML$_{soft}$	$\mathbf{64.27}_{\pm \mathbf{0.75}}$	$\mathbf{54.73}_{\pm \mathbf{0.75}}$	$\mathbf{46.91}_{\pm \mathbf{0.71}}$	$\underline{61.47_{\pm 0.71}}$	$\mathbf{56.84}_{\pm \mathbf{0.73}}$
UAPML$_{hard}$	$\underline{63.12_{\pm 0.75}}$	$\underline{54.11_{\pm 0.73}}$	$\underline{46.85_{\pm 0.73}}$	$61.34_{\pm 0.72}$	$\underline{56.36_{\pm 0.73}}$

Meta-SGD (Li et al., 2017): an efficient method that learns the appropriate learning rate to enable the one-step adaption to few-shot tasks.

Sparse-MAML (Von Oswald et al., 2021): a state-of-the-art efficient method that aims to learn which weight to change during inner adaptation, allowing apart of parameters to adapt to tasks. It employs an additional modulator with the same scale as the model architecture to determine where to update during inner adaptation.

LEO (Rusu et al., 2019): a conditional meta-learning method that captures task information and operates the inner adaptation in a latent space. It then uses the latent vector to generate a task-specific task head to deal with the specific task.

Sharp-MAML (Abbas et al., 2022): a state-of-the-art variant of MAML that employs sharpness-aware minimization tackle the complex loss landscape caused by the bi-level optimization, achieving a better performance.

A.4.3 The Details of the Experimental settings

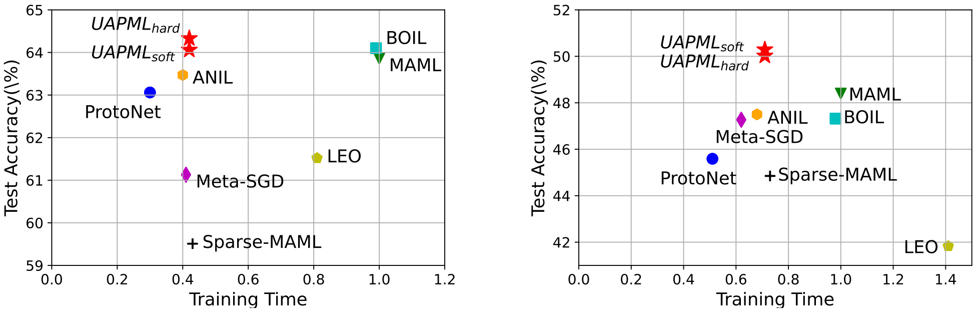
We employ the same architecture as the base model in MAML (Finn et al., 2017) for all the baselines and our proposed model. The architecture comprises 4 convolution layers with 64 filters, batch normalization, a ReLU non-linear layer, and a 2×2 max-pool. A larger architecture ResNet12 is also used to examinate the effectiveness and efficiency of all methpds. Except for the conventional methods (i.e., MAML, ANIL, and BOIL), all the baselines are based on the public repositories released by the original papers. In our proposed UAPML, we use the full-connect layer as the merge module, which is the simplest layer. We set the tuning step as 4 and the tuning rate as 0.01 in UAPML. Moreover, we use Adam optimization with a learning rate of 0.001 to learn the meta-prompt and other meta-parameters. All models are trained on NVIDIA A40 for 10000 epochs, with each epoch consisting of a batch of 4 tasks from different task distributions. The training process takes less than one hour under both 5-shot and 1-shot settings.

Table 6: **5-way 5-shot** accuracy (%) with 95% confidence interval of ResNet12 architecture on each task distribution and on average. The best and the second-best performances are bolded and underlined, respectively.

	Aircraft	CIFAR-FS	Mini-Imagenet	miniQuickDraw	Average
MAML	$76.46_{\pm 0.58}$	$72.04_{\pm 0.62}$	$62.08_{\pm 0.68}$	$79.91_{\pm 0.51}$	$72.62_{\pm 0.60}$
ProtoNet	$76.35_{\pm 0.50}$	$71.60_{\pm 0.59}$	$61.91_{\pm 0.61}$	$\mathbf{81.25_{\pm 0.46}}$	$72.78_{\pm 0.54}$
ANIL	$76.87_{\pm 0.59}$	$72.32_{\pm 0.60}$	$62.38_{\pm 0.66}$	$77.77_{\pm 0.58}$	$72.33_{\pm 0.61}$
BOIL	$32.85_{\pm 0.51}$	$49.38_{\pm 0.60}$	$45.49_{\pm 0.60}$	$64.31_{\pm 0.58}$	$48.01_{\pm 0.57}$
Meta-SGD	$60.89_{\pm 0.56}$	$51.27_{\pm 0.58}$	$46.48_{\pm 0.62}$	$56.87_{\pm 0.60}$	$53.88_{\pm 0.59}$
LEO	$76.30_{\pm 0.55}$	$72.06_{\pm 0.57}$	$63.86_{\pm 0.62}$	$79.85_{\pm 0.47}$	$73.02_{\pm 0.55}$
Sparse-MAML	$77.79_{\pm 0.60}$	$\mathbf{75.18_{\pm 0.63}}$	$63.32_{\pm 0.70}$	$76.70_{\pm 0.62}$	$73.25_{\pm 0.64}$
Sharp-MAML	$77.37_{\pm 0.56}$	$74.15_{\pm 0.60}$	$63.17_{\pm 0.67}$	$79.27_{\pm 0.50}$	$73.49_{\pm 0.58}$
UAPML$_{soft}$	$\mathbf{80.57_{\pm 0.51}}$	$73.66_{\pm 0.60}$	$62.74_{\pm 0.68}$	$78.11_{\pm 0.51}$	$73.77_{\pm 0.57}$
UAPML$_{hard}$	$78.32_{\pm 0.51}$	$74.82_{\pm 0.60}$	$\mathbf{64.68_{\pm 0.65}}$	$79.20_{\pm 0.49}$	$\mathbf{74.25_{\pm 0.56}}$

Table 7: **5-way 1-shot** accuracy (%) with 95% confidence interval of ResNet12 architecture on each task distribution and on average. The best and the second-best performances are bolded and underlined, respectively.

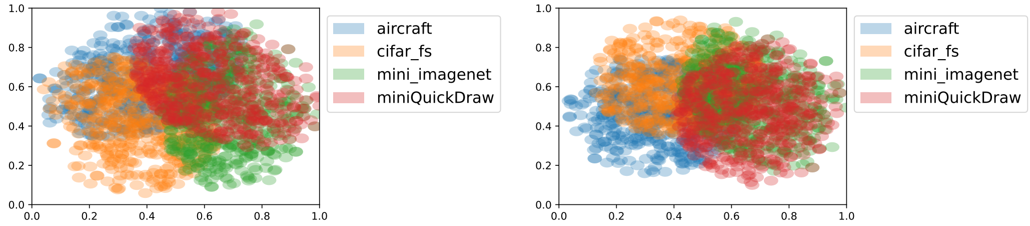
	Aircraft	CIFAR-FS	Mini-Imagenet	miniQuickDraw	Average
MAML	$60.26_{\pm 0.73}$	$53.86_{\pm 0.78}$	$44.31_{\pm 0.71}$	$62.02_{\pm 0.75}$	$55.11_{\pm 0.74}$
ProtoNet	$60.56_{\pm 0.71}$	$53.14_{\pm 0.70}$	$46.26_{\pm 0.67}$	$59.31_{\pm 0.70}$	$54.82_{\pm 0.69}$
ANIL	$59.06_{\pm 0.71}$	$52.26_{\pm 0.78}$	$43.79_{\pm 0.70}$	$62.63_{\pm 0.78}$	$54.44_{\pm 0.74}$
BOIL	$26.61_{\pm 0.47}$	$35.44_{\pm 0.61}$	$32.02_{\pm 0.53}$	$42.93_{\pm 0.65}$	$34.25_{\pm 0.57}$
Meta-SGD	$27.98_{\pm 0.45}$	$38.23_{\pm 0.64}$	$35.56_{\pm 0.62}$	$45.40_{\pm 0.71}$	$36.80_{\pm 0.60}$
LEO	$30.64_{\pm 0.50}$	$42.62_{\pm 0.69}$	$37.87_{\pm 0.65}$	$55.71_{\pm 0.73}$	$41.71_{\pm 0.64}$
Sparse-MAML	$62.48_{\pm 0.77}$	$55.98_{\pm 0.82}$	$47.01_{\pm 0.79}$	$62.00_{\pm 0.81}$	$56.87_{\pm 0.80}$
Sharp-MAML	$62.15_{\pm 0.76}$	$54.84_{\pm 0.80}$	$44.70_{\pm 0.69}$	$\mathbf{64.70_{\pm 0.79}}$	$56.60_{\pm 0.76}$
UAPML$_{soft}$	$\mathbf{64.92_{\pm 0.77}}$	$\mathbf{57.94_{\pm 0.79}}$	$\mathbf{47.05_{\pm 0.73}}$	$64.15_{\pm 0.78}$	$\mathbf{58.52_{\pm 0.77}}$
UAPML$_{hard}$	$63.18_{\pm 0.81}$	$55.94_{\pm 0.78}$	$46.76_{\pm 0.73}$	$64.36_{\pm 0.74}$	$57.56_{\pm 0.77}$



(a) 5-way 5-shot on Mini-Imagenet

(b) 5-way 1-shot on Mini-Imagenet

Figure 5: The 5-way 5-shot and 5-way 1-shot accuracy and training time on Mini-Imagenet.



(a) UAPML$_{soft}$

(b) UAPML$_{hard}$

Figure 6: The visualization for task-specific prompts of UAPML$_{soft}$ and UAPML$_{hard}$ under 5-way 5-shot setting.

Table 8: **5-way 5-shot** accuracy (%) with 95% confidence interval of the variants of our UAPML on each task distribution and on average.

		Aircraft	CIFAR-FS	Mini-Imagenet	miniQuickDraw	Average
UAPML$_{soft}$	dim=32	76.10$_{\pm0.53}$	71.75$_{\pm0.57}$	61.51$_{\pm0.68}$	77.14$_{\pm0.54}$	71.62$_{\pm0.58}$
	dim=64	76.32$_{\pm0.54}$	72.37$_{\pm0.60}$	61.83$_{\pm0.65}$	77.31$_{\pm0.55}$	71.96$_{\pm0.59}$
	dim=128	74.84$_{\pm0.54}$	71.65$_{\pm0.59}$	61.61$_{\pm0.66}$	77.73$_{\pm0.53}$	71.46$_{\pm0.58}$
	dim=256	74.19$_{\pm0.55}$	71.80$_{\pm0.60}$	61.42$_{\pm0.66}$	77.19$_{\pm0.53}$	71.15$_{\pm0.59}$
	dim=512	77.03$_{\pm0.51}$	70.92$_{\pm0.59}$	60.60$_{\pm0.66}$	76.68$_{\pm0.53}$	71.31$_{\pm0.57}$
	kl=0.01	75.13$_{\pm0.57}$	71.71$_{\pm0.59}$	61.50$_{\pm0.67}$	77.67$_{\pm0.51}$	71.50$_{\pm0.59}$
	kl=5	74.26$_{\pm0.55}$	70.73$_{\pm0.59}$	61.26$_{\pm0.65}$	77.25$_{\pm0.54}$	70.88$_{\pm0.58}$
	kl=10	73.92$_{\pm0.56}$	71.41$_{\pm0.59}$	61.21$_{\pm0.64}$	77.35$_{\pm0.51}$	70.97$_{\pm0.58}$
UAPML$_{hard}$	dim=32	75.21$_{\pm0.54}$	72.01$_{\pm0.58}$	61.59$_{\pm0.67}$	76.78$_{\pm0.54}$	71.40$_{\pm0.58}$
	dim=64	76.53$_{\pm0.56}$	71.48$_{\pm0.56}$	61.59$_{\pm0.65}$	77.63$_{\pm0.52}$	71.81$_{\pm0.57}$
	dim=128	75.28$_{\pm0.54}$	72.23$_{\pm0.59}$	60.92$_{\pm0.64}$	76.91$_{\pm0.52}$	71.34$_{\pm0.57}$
	dim=256	75.24$_{\pm0.56}$	70.93$_{\pm0.62}$	61.32$_{\pm0.68}$	76.57$_{\pm0.54}$	71.02$_{\pm0.60}$
	dim=512	75.17$_{\pm0.57}$	70.20$_{\pm0.63}$	60.82$_{\pm0.64}$	76.45$_{\pm0.52}$	70.66$_{\pm0.59}$
	kl=0.01	76.04$_{\pm0.52}$	72.34$_{\pm0.59}$	62.60$_{\pm0.65}$	77.81$_{\pm0.54}$	72.20$_{\pm0.57}$
	kl=5	76.15$_{\pm0.56}$	72.56$_{\pm0.59}$	62.64$_{\pm0.66}$	78.41$_{\pm0.51}$	72.44$_{\pm0.58}$
	kl=10	75.94$_{\pm0.56}$	70.61$_{\pm0.57}$	61.82$_{\pm0.66}$	76.91$_{\pm0.53}$	71.32$_{\pm0.58}$
N/A	w/o Bayes	74.41$_{\pm0.57}$	69.93$_{\pm0.61}$	60.83$_{\pm0.68}$	76.93$_{\pm0.54}$	70.53$_{\pm0.60}$
N/A	w/o C	75.57$_{\pm0.55}$	70.87$_{\pm0.59}$	60.93$_{\pm0.65}$	77.14$_{\pm0.54}$	71.13$_{\pm0.58}$

Table 9: **5-way 1-shot** accuracy (%) with 95% confidence interval of the variants of our UAPML on each task distribution and on average.

		Aircraft	CIFAR-FS	Mini-Imagenet	miniQuickDraw	Average
UAPML$_{soft}$	dim=32	63.77$_{\pm0.72}$	53.25$_{\pm0.76}$	46.33$_{\pm0.75}$	61.75$_{\pm0.73}$	56.28$_{\pm0.74}$
	dim=64	64.27$_{\pm0.75}$	54.73$_{\pm0.75}$	46.91$_{\pm0.71}$	61.47$_{\pm0.71}$	56.84$_{\pm0.73}$
	dim=128	63.41$_{\pm0.74}$	53.94$_{\pm0.75}$	46.21$_{\pm0.75}$	62.36$_{\pm0.73}$	56.48$_{\pm0.74}$
	dim=256	62.88$_{\pm0.74}$	54.69$_{\pm0.74}$	47.23$_{\pm0.72}$	61.79$_{\pm0.76}$	56.65$_{\pm0.74}$
	dim=512	62.90$_{\pm0.68}$	53.82$_{\pm0.76}$	46.57$_{\pm0.74}$	61.93$_{\pm0.70}$	56.31$_{\pm0.72}$
	kl=0.01	62.83$_{\pm0.72}$	52.94$_{\pm0.73}$	47.09$_{\pm0.72}$	60.98$_{\pm0.72}$	55.96$_{\pm0.72}$
	kl=5	63.21$_{\pm0.76}$	55.37$_{\pm0.77}$	45.32$_{\pm0.70}$	61.00$_{\pm0.73}$	56.23$_{\pm0.74}$
	kl=10	62.80$_{\pm0.74}$	53.49$_{\pm0.77}$	46.36$_{\pm0.71}$	62.10$_{\pm0.75}$	56.19$_{\pm0.74}$
UAPML$_{hard}$	dim=32	63.61$_{\pm0.75}$	53.47$_{\pm0.76}$	46.08$_{\pm0.68}$	61.94$_{\pm0.73}$	56.28$_{\pm0.73}$
	dim=64	63.12$_{\pm0.75}$	54.11$_{\pm0.73}$	46.85$_{\pm0.73}$	61.34$_{\pm0.72}$	56.36$_{\pm0.73}$
	dim=128	63.54$_{\pm0.75}$	53.58$_{\pm0.77}$	46.54$_{\pm0.71}$	60.73$_{\pm0.71}$	56.10$_{\pm0.74}$
	dim=256	63.51$_{\pm0.74}$	53.77$_{\pm0.73}$	47.09$_{\pm0.73}$	60.83$_{\pm0.77}$	56.30$_{\pm0.74}$
	dim=512	63.10$_{\pm0.71}$	54.00$_{\pm0.75}$	46.00$_{\pm0.70}$	61.74$_{\pm0.74}$	56.21$_{\pm0.73}$
	kl=0.01	63.51$_{\pm0.73}$	53.38$_{\pm0.75}$	46.04$_{\pm0.70}$	61.71$_{\pm0.75}$	61.71$_{\pm0.75}$
	kl=5	65.87$_{\pm0.73}$	53.58$_{\pm0.72}$	46.53$_{\pm0.72}$	61.67$_{\pm0.71}$	56.91$_{\pm0.72}$
	kl=10	63.42$_{\pm0.74}$	52.60$_{\pm0.72}$	46.98$_{\pm0.71}$	62.62$_{\pm0.74}$	56.41$_{\pm0.73}$
N/A	w/o Bayes	62.07$_{\pm0.76}$	53.07$_{\pm0.75}$	45.27$_{\pm0.72}$	61.47$_{\pm0.73}$	55.47$_{\pm0.74}$
N/A	w/o C	63.63$_{\pm0.70}$	52.54$_{\pm0.77}$	46.24$_{\pm0.73}$	60.89$_{\pm0.73}$	55.82$_{\pm0.73}$

A.5 The Experimental Result

At first, we showed the detailed results of all methods of Conv4 architecture on the mixture datasets in Tab. 4 and Tab. 5, including the average performance and the respective performance on each dataset. The detailed results demonstrate that under both the 5-shot and 1-shot setting, UAPML achieve a statistically significant improvement on all datasets except *miniQuickDraw* according to Student's t-test (Sakai, 2016). This further confirms the effectiveness of our proposed UAPML. with regards to the architecture ResNet12, we also show the 5-way 5-shot and 5-way 1-shot results in Tab. 6 and Tab. 7, respectively. Although some baselines perform best on a certain single dataset, our proposed UAPML still has a significant improvement in average accuracy.

Moreover, we also report the performance of all methods on the vanilla dataset Mini-Imagenet in Fig. 5, including the test accuracy and the time consumption for training. Even on this vanilla dataset, our proposed UAPML achieves the best performance under 5-shot and 1-shot settings. Although the increase on 5-shot is not significant, our proposed UAPML only uses 40% time for training when compared to MAML. The results illustrate that our proposed UAPML is a more effective and efficient meta-learning method.

Furthermore, we show the detailed performance of the variant of UAPML, i.e., the performance on each task distribution, in this subsection. The detailed results under 5-shot and 1-shot settings are shown in Tab. 8 and Tab. 9, respectively.

We also visualize the task-specific prompt under the 5-shot setting. It is observed that task-specific prompts for each dataset are separated but still have some overlap. This is because the construction for task-specific prompts limits or even freezes part of the dimensions when adating oto tasks.