Toward a More Discriminative Learnware Paradigm via Explicitly Distinctive Specification

Wenlu Yang[1] , **Wei Chen**[2,3] and **Zhenan He**[1*]
[1]College of Computer Science, Sichuan University
[2]School of Computer Science and Engineering, Southeast University, Nanjing, China
[3]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
yangwenlu0909@gmail.com, wei-chen@seu.edu.cn, zhenan@scu.edu.cn

Abstract

The *learnware paradigm* aims to construct a dock system that maintains a collection of learnwares, each consisting of a well-established model coupled with a *specification*, thereby enabling users to directly reuse existing models to solve their tasks without training models from scratch. As a core component of the paradigm, the specification sketches the model's properties and establishes reusability between the learnwares and user tasks. Existing methods have demonstrated promising performance by designing specifications that characterize the task distributions mastered by well-established models. However, in complex real-world scenarios, task distributions often overlap, thereby weakening the uniqueness of specifications and obscuring the reusability relationships between learnwares and tasks. In this paper, we design an ***E**xplicitly **D**istinctive **S**pecification* (EDS) that enforces specification uniqueness, improving the discriminative capability of the learnware paradigm and avoiding ambiguity caused by overlapping task distributions. This specification method explicitly leverages distributional discrepancies between specifications to strengthen the uniqueness of the model properties sketched during the submitting stage, and subsequently exploits the enhanced distributional discriminability in the deploying stage to improve learnware reusability. Extensive experiments demonstrate the effectiveness and strong performance of the proposed EDS within the learnware paradigm under complex task scenarios.

1 Introduction

Recent advances in machine learning, along with its expanding real-world applications, have led to the proliferation of diverse models [Jordan and Mitchell, 2015]. However, under the conventional machine learning paradigm, models are typically constructed from scratch for each task, a process that entails substantial resource demands [Ye *et al.*, 2021]. Specifically, effective model development often requires the simultaneous availability of large-scale labeled datasets [Anaby-Tavor *et al.*, 2020], high-performance computational infrastructure [Shazeer and et al., 2018], and extensive domain expertise [von Rueden *et al.*, 2019], which together significantly raise the barrier to practical deployment. Moreover, constraints imposed by data privacy and proprietary considerations further hinder model development and reuse across tasks [Shejwalkar and Houmansadr, 2021].

The *learnware paradigm* [Zhou, 2016] provides a systematic approach to addressing these intertwined challenges by introducing a *learnware dock system* that aggregates reusable learnwares, each consisting of a well-established model and a corresponding *specification* describing its specialty. Through specification-guided retrieval, users can directly reuse existing models to solve diverse tasks without retraining. The learnware paradigm consists of two stages: (1) *the submitting stage*, during which developers upload well-established models, and the dock system automatically assigns appropriate specifications to form learnwares; and (2) *the deploying stage*, in which the dock system searches for specifications that best match users' specific tasks and subsequently reuses the corresponding learnware models to solve those tasks.

Therefore, it is evident that specification generation lies at the core of the submitting stage, as it sketches the task distribution that the model has mastered and establishes reusability relationships between learnwares and user tasks. Existing methods primarily construct specifications via distribution alignment, using limited data to approximate the task distribution learned by each model. While effective at preserving distributional consistency, these approaches treat specifications independently and ignore inter-specification relationships, leading to overlapping descriptions, reduced discriminability, and ambiguous model–task matching.

Furthermore, the specification is equally critical in the deploying stage. Under the *task recurrence assumption* [Wu *et al.*, 2023], learnware retrieval can be achieved through direct specification matching. Under the more general *instance recurrence assumption* [Wu *et al.*, 2023], instance-level model selection becomes necessary, typically by training a learnware index selector on mimic samples generated to approximate user tasks. However, existing methods usually generate such samples at a coarse task level based on specifications, overlooking the intrinsic multimodal structure of real-world data. This limitation results in insufficient sample diversity,

*Corresponding authors.

weakens the selector's generalization ability, and ultimately degrades deployment performance.

Overall, existing methods lack explicit discriminative considerations at the submitting stage and suffer from insufficient diversity in mimic data generation during the deploying stage. To address these limitations, we propose an ***Explicitly Distinctive Specification*** (EDS), which explicitly enforces specification uniqueness during the submitting stage. Specifically, EDS incorporates a discriminative penalty that enlarges the margin between newly generated specifications and existing ones, thereby enhancing the discriminative capability of the learnware paradigm and alleviating ambiguous identification of user-relevant learnwares caused by specification overlap. Building upon these discriminability-enhanced specifications, we generate mimic data during the deploying stage by exploiting their class-level structural information and local statistical characteristics adaptively, ensuring alignment with user requirements while enhancing sample diversity. Extensive experiments on both synthetic and real-world datasets demonstrate that EDS consistently improves learnware identification accuracy and model reuse performance in the learnware paradigm.

2 Related Works

The learnware paradigm [Zhou, 2016] enables systematic model reuse by solving tasks through existing learnware, instead of designing and training models from scratch. This effectively alleviates several critical bottlenecks in practical artificial intelligence deployment, including resource constraints, data privacy concerns, and the burden of repetitive model development [Zhou and Tan, 2024]. In this paradigm, learnware incorporates a well-established model and a specification that sketches its capabilities. Hence, the key to systematically managing and reusing learnware lies in these specifications [Tan *et al.*, 2024c].

As a cornerstone of the learnware paradigm, specification generation in the submitting stage has become a central research focus, aiming to construct representations that effectively characterize the distinctive capabilities of individual models [Chen *et al.*, 2025b]. Early work introduced Reduced Kernel Mean Embeddings (RKME) to generate specifications by approximating task data distributions in the RKHS [Wu *et al.*, 2023], with subsequent theoretical analysis demonstrating its strong privacy-preserving properties [Lei *et al.*, 2024]. Building on RKME, a series of studies have extended specification generation to increasingly challenging scenarios, including unseen user tasks [Zhang *et al.*, 2021], heterogeneous feature spaces [Tan *et al.*, 2024b; Tan *et al.*, 2023; Tan *et al.*, 2024a], and heterogeneous label spaces [Guo *et al.*, 2023]. In addition, Chen *et al.*[2025b] replaced kernel mean embeddings with random neural embeddings, alleviating the strong dependence of RKME-based methods on kernel selection. By further exploiting models' latent discriminative representations, random neural embeddings enable more precise specifications and naturally support heterogeneous label scenarios [Chen *et al.*, 2025a]. However, existing specification approaches mainly emphasize intra-specification distribution alignment, while largely overlooking inter-specification discriminability. In real-world scenarios with overlapping task distributions, this limitation leads to blurred specification boundaries and ambiguous learnware–task associations, thereby impairing effective model identification and reuse.

As for the deploying stage, since it pertains to the application of generating specifications, few methods focus solely on this stage [Xie *et al.*, 2023]. Most existing approaches adopt RKME's kernel herding strategy to produce synthetic data that approximates user data. However, the reliance on kernel functions imposes limitations and significant computational costs. In contrast, Chen *et al.*[2025b] generate mimic data using a multivariate normal distribution, avoiding the challenges associated with kernel selection. Nevertheless, the task-level statistical generation constrains sample diversity, undermining the generalization of learnware selectors.

To overcome the limited discriminability of specifications at the submitting stage and the insufficient diversity of mimic data during deploying stage, we introduce an *Explicitly Distinctive Specification* (EDS) that enhances the uniqueness of learnwares in the submitting stage and further investigates their internal characteristics to increase the diversity of mimic data within the deploying stage.

3 Preliminary

The learnware paradigm is organized into two key stages, namely the submitting stage and the deploying stage.

The submitting stage. During this stage, each developer $i \in [c]$ can autonomously submit the learnware dock system a well-established model f_i trained on private dataset $\mathcal{D}_i = \{(\boldsymbol{x}_{i,n}, y_{i,n})\}_{n=1}^{N_i}$ corresponding task $T_i := \{P_i, f_i\}, i \in \{1, ..., c\}$, where P_i is a distribution based on the input space $\mathcal{X}$. $f_i : \mathcal{X} \to \mathcal{Y}$ is a well-established model that transforms $\mathcal{X}$ to the output space $\mathcal{Y}$, and satisfies

$$\begin{aligned} &\forall(\boldsymbol{x}_{i,n}, y_{i,n}) \in D_i, f(\boldsymbol{x}_{i,n}) = y_{i,n}, \\ &s.t. \quad \forall i \in \{i, \cdots, c\}, \forall n \in [N_i]. \end{aligned} \tag{1}$$

Then, the dock system generates a specification Φ approximating of the model's mastery distribution to characterize its capabilities, expressed as:

$$\left\|\mu(P_X) - \mu(P_\Phi)\right\|^2, \tag{2}$$

where $\mu(\cdot)$ denotes a distribution function that characterizes the underlying distributional pattern. Currently, the mainstream RKME method characterizes distributions via kernel mean embedding, whereas the LANE method employs random neural embedding for distribution characterization. Detailed descriptions of these methods are provided in supplementary material A and B. Finally, the specification Φ_i and its corresponding well-established model f_i jointly constitute a learnware (f_i, Φ_i), which is then stored in the dock system to facilitate subsequent reuse.

The deploying stage. In this stage, a user aims to solve a requirement task $\hat{T} := \{P_{\hat{X}}, \hat{f}\}$ with, that is, to obtain a model f capable of addressing $\hat{T}$ by minimizing $\mathcal{L}(P_{\hat{X}}, \hat{f}, f) \triangleq \mathbb{E}_{\hat{\boldsymbol{x}} \sim P_{\hat{X}}}[L(f(\hat{\boldsymbol{x}}), \hat{f}(\hat{\boldsymbol{x}}))]$ with the assistance of the learnware

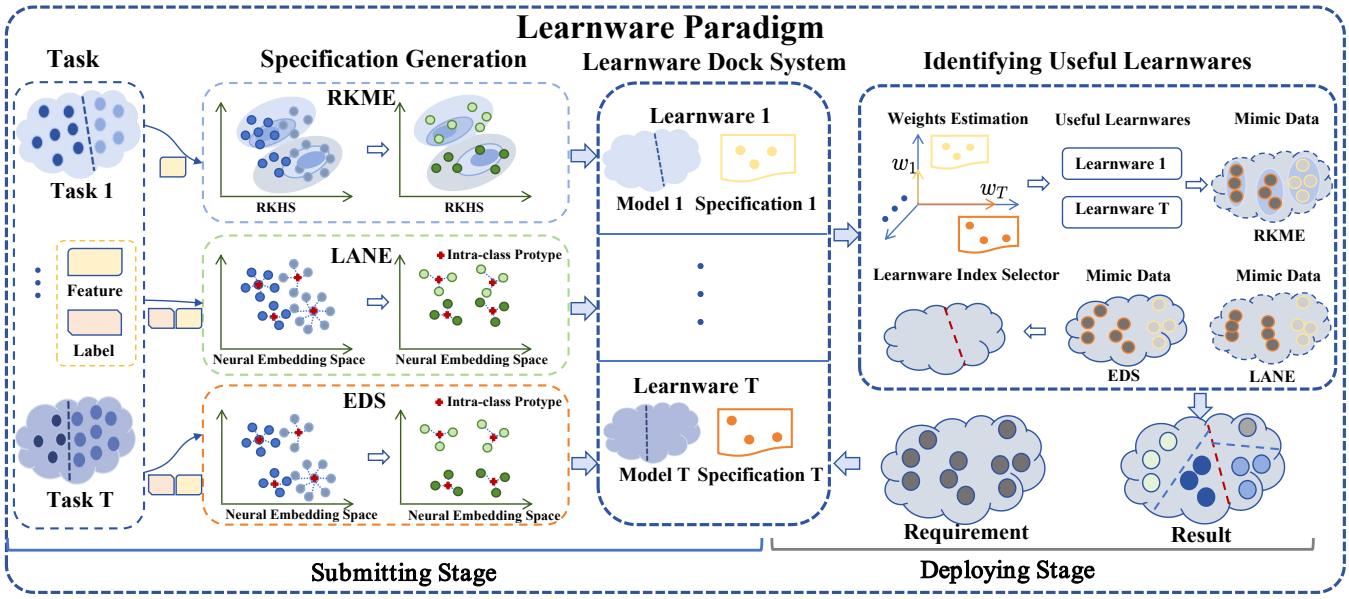

Figure 1: The learnware paradigm of EDS, LANE and RKME. Notably, in the submission phase, compared with RKME and LANE, the light and dark green representations generated by EDS for different tasks exhibit significantly greater discriminability. Furthermore, during the deployment stage, the mimic samples generated by EDS are more widely dispersed, indicating enhanced diversity over the other two methods. The enhanced discriminability of specifications, combined with the increased diversity of mimic samples, collectively empowers EDS to attains superior performance in achieving accurate and efficient model reuse.

dock system. Specifically, the paradigm stipulates that the dock system identifies useful learnwares capable of addressing user tasks based on existing specifications and subsequently returns them to the user. Formally speaking, the user task $\hat{T}$ has a distribution relationship with the specifications in the dock system, expressed as $P_{\hat{X}} = \sum_{i=1}^{c} \boldsymbol{w}_i P_{X_i}$, representing the reusability relationship between the task and the learnware. Hence, the process is formulated as follows:

$$\min_{\boldsymbol{w}} \left\| \mu(P_{\hat{X}}) - \sum_{i=1}^{c} w_i \mu(P_{\Phi_i}) \right\|^2, \quad s.t.\ w_i \geq 0, \forall i \in [c] \ and \ \sum_{i=1}^{c} w_i = 1. \tag{3}$$

where Φ_i denotes the i-th learnware specification in the learnware dock system, and w_i indicates the degree of usefulness of the i-th learnware model for the user task. Then, to handle tasks without background information (i.e., label information), mimic data X^{mimic} is generated according to Eq.(3) to ensure that the mixture of useful specification distributions can effectively approximate the task distribution. Its formulation is as follows:

$$X^{mimic} = \{(\boldsymbol{x}_j^{mimic}, i_j)\} \sim \sum w_i P_{\Phi_i}, \tag{4}$$

where X^{mimic} is the sample set with learnware index label. In learnware paradigms based on the RKME and LANE methods, this process adopts different distribution sampling strategies corresponding to their respective distribution characterizations, as detailed in supplementary material A and B. Finally, by training a learnware index selector $g(\cdot)$ (e.g., SVM) on mimic data X^{mimic}, useful learnware models can be matched to appropriate task portions, thereby enabling the effective reuse of well-established models to solve user tasks.

4 Methodology

Existing specification methods are primarily designed to achieve distributional alignment, with limited consideration of discriminative separability among specifications. Indeed, in complex real-world scenarios, task distributions often exhibit substantial overlap. This oversight weakens specification uniqueness and obscures the reusability relationships between learnwares and user tasks. To overcome this limitation, we propose an **E***xplicitly* **D***istinctive* **S***pecification* (EDS) method, as illustrated in Fig. 1. EDS explicitly exploits distributional discrepancies among specifications to enforce the uniqueness of the model properties sketched during the submitting stage (EDS-S), and further exploits the enhanced distributional discriminability in the deploying stage (EDS-D) to improve learnware identification and reuse.

4.1 The Submitting Stage

At this stage, we argue that, beyond distribution alignment, enhancing discriminability among specifications is essential for achieving clear specification uniqueness. This mitigates ambiguity caused by overlapping distributions and enables more accurate identification of user-relevant requirements in subsequent stages. Accordingly, the specification generation process is guided by two objectives: (1) *intra-specification distribution preservation*, which ensures that each specification approximates its training data distribution at the class level rather than over the entire dataset; (2) *inter-specification*

discrimination, which explicitly exploits distributional discrepancies among specifications to reinforce the uniqueness of the model properties they sketch.

Intra-specification distribution preservation. To sketch well-established model properties and establish reusability relationships between learnwares and tasks, the specification is designed to characterize the distribution mastered by the model. Unlike RKME, which matches the full training data distribution in a reproducing kernel Hilbert space (RKHS), the proposed (EDS-S) employs random neural embedding to achieve alignment at the class level, thereby maintaining distributional fidelity while enhancing structural semantics. Assume that the dataset $\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$ comprises K categories, where $\mathcal{X} = \mathbb{R}^d (\mathrm{x} \in \mathcal{X})$ denotes samples in the feature space and $\mathcal{Y} = \mathbb{R}^K (y \in \mathcal{Y})$ indicates their associated labels in the label space. Then, for each class k, the method computes the mean representations of samples from the original dataset $\mathcal{D}$ and the specification $\Phi = \{(\boldsymbol{z}_m, y_m)\}_{m=1}^{M}$ corresponding to the k-th class in the embedding space $\psi_{\boldsymbol{v}}$ [Zhao and Bilen, 2021]. By minimizing their discrepancy, the model drives Φ to preserve the intra-class distributional structure of $\mathcal{D}$, thereby achieving distribution-preserving compression that maintains both statistical fidelity and structural integrity. The aforementioned process can be formalized as follows:

$$\mathcal{L}_{\mathrm{MMD}} = \mathbb{E}_{\boldsymbol{v}\sim P(\boldsymbol{v})} \sum_{k=1}^{K} \left\| \frac{1}{|\mathcal{D}_k|} \sum_{\boldsymbol{x}\in\mathcal{D}_k} \psi_{\boldsymbol{v}}(\boldsymbol{x}) - \frac{1}{|\Phi_k|} \sum_{\boldsymbol{z}\in\Phi_k} \psi_{\boldsymbol{v}}(\boldsymbol{z}) \right\|^2, \tag{5}$$

where the random initialize neural network $\psi_{\boldsymbol{v}} \colon \mathbb{R}^d \to \mathbb{R}^{d'}$ maps high-dimensional inputs into a lower-dimensional latent space ($d' \ll d$) with parameters $\boldsymbol{v} \sim P_{\boldsymbol{v}}$.

Benefiting from random network $\psi_{\boldsymbol{v}}$, as stated in [Dong *et al.*, 2022], the above approach is able to represent the dataset $\mathcal{D}$ using a small amount of data, while ensuring that the class-level center of gravity of the $\mathcal{D}$ remains consistent with that of the generated specifications Φ.

Inter-specification discrimination. Existing specification methods primarily aim to minimize the distributional gap between the specification and the task distribution mastered by the model, thereby ensuring statistical consistency. However, due to the high complexity of real-world tasks, neglecting discriminability among specifications can blur their boundaries, resulting in insufficient distinctiveness and separation. This deficiency weakens the ability of specifications to faithfully represent the underlying models and introduces ambiguity in subsequent model retrieval. Since each specification uniquely corresponds to a learnware model in the learnware paradigm, overlapping or poorly separated specifications may lead to ambiguous model selection for boundary instances, whereas well-separated specifications enable more decisive and reliable model identification.

To alleviate this issue, a discriminative regularization term is introduced to explicitly generate distinctive specifications. The core idea is to impose a "repulsive force" within the embedding space, ensuring that each specification maintains an independent feature orientation and a reasonable separation from others. Formally, let the current specification data be denoted as $\{\boldsymbol{z}_i\}_{i=1}^{M}$ and the previously obtained as $\{\boldsymbol{z}'_j\}_{j=1}^{M}$. The discrimination regularization term is then defined as:

$$\mathcal{L}_{\text{dicri}} = \frac{1}{N} \sum_{i=1}^{N} \max\left(0, \cos(\boldsymbol{z}_i, \boldsymbol{z}'_j) - \delta\right), \tag{6}$$

$$\cos(\boldsymbol{z}_i, \boldsymbol{z}'_j) = \frac{\boldsymbol{z}_i^\top \boldsymbol{z}'_j}{\|\boldsymbol{z}_i\|_2 \, \|\boldsymbol{z}'_j\|_2}. \tag{7}$$

Here, $\cos(\boldsymbol{z}_i, \boldsymbol{z}'_j)$ denotes the cosine similarity between two specification representations, and δ represents the similarity margin that controls the allowable proximity between them. By explicitly incorporating $\mathcal{L}_{\text{dicri}}$, the proposed (EDS-S) is guided toward generating specifications that exhibit both distinctiveness and diversity.

Overall. By integrating the intra-specification distribution alignment and inter-specification discrimination, the overall objective of EDS-S stage can be formulated as:

$$\min_{\boldsymbol{z}} \mathcal{L}_{\text{MMD}} + \mathcal{L}_{\text{dicri}}. \tag{8}$$

Consequently, by preserving class-level data distribution through $\mathcal{L}_{\text{MMD}}$ and enhancing inter-specification distinctiveness through $\mathcal{L}_{\text{dicri}}$, the proposed method yields more accurate discriminative learnware, facilitating reliable identification and reuse of learnware models. Based on Proposition 4.3 in [Chen *et al.*, 2025a] and Theorem 1 in [Mohri and Muñoz Medina, 2012], a theoretical upper bound for $\mathcal{L}_{\text{discri}}$ can be derived, as detailed in Appendix D, while the submitting procedure is provided in Algorithm 1 of Appendix C. In the following subsection, we describe how users leverage the learnware dock system to solve their own tasks using the existing learnwares.

4.2 The Deploying Stage

During this stage, users submit their tasks to the learnware dock system, which retrieves the learnware that best matches user requirements without any background knowledge, enabling the reuse of either a single well-established model or an appropriately selected ensemble of models.

Specifically, mixture weight estimation is first performed to identify task-relevant learnwares. Subsequently, to address task requirements without any prior background knowledge and maintain the privacy-preserving property of the learnware paradigm, a mimic dataset resembling the user requirements is generated from these task-relevant learnwares, which is then used to train a learnware index selector to choose appropriate learnware models for reuse. As the basis for training the selector, the diversity of mimic data directly affects the selector's generalization ability and ultimately determines the choice of reusable models. Building on this, in this deploying stage, our goal is to go beyond the intrinsic diversity naturally provided by the specifications and additionally leverage their class-level characteristics to further enhance the diversity of the mimic data, thereby enabling more effective instance-wise model selection.

Assume that the user requirement is $\hat{\mathcal{D}} = \{(\hat{\boldsymbol{x}}_{\hat{n}})\}_{\hat{n}=1}^{\hat{N}}$, the specification generated based on this data is denoted as $\hat{\Phi} = \{\hat{\boldsymbol{z}}_{\boldsymbol{i}}\}_{i=1}^{M}$. The system subsequently determines the most compatible learnware by quantifying the correlation between the requirement and existing learnware specifications, formulated as follows.

$$\min_{\boldsymbol{w}} \quad \left\| \frac{1}{\hat{N}} \sum_{\hat{n}=1}^{\hat{N}} \psi_{\boldsymbol{v}}(\hat{\boldsymbol{x}}_{\hat{n}}) - \sum_{i=1}^{c} \sum_{k=1}^{K_i} \frac{w_{i,k}}{|\Phi_{i,k}|} \sum_{\boldsymbol{z} \in \Phi_{i,k}} \psi_{\boldsymbol{v}}(\boldsymbol{z}) \right\|$$
$$\text{s.t.} \quad w_{i,k} \geq 0, \forall i \in [c], k \in [K] and \sum_{i=1}^{c} \sum_{k=1}^{K_i} w_{i,k} = 1. \tag{9}$$

Unlike previous approaches that estimate relevance only at the task level, the proposed deploying stage based on EDS (hereafter EDS-D) performs correlation estimation at the class level within each task, enabling a more fine-grained decomposition of user requirements. Accordingly, $\Phi_{i,k}$ denotes the specification of the k-th class in the i-th learnware, while $w_{i,k}$ represents the correlation between the requirement and $\Phi_{i,k}$. Following the above definitions, the weights satisfy $\sum_{i=1}^{c} \sum_{k=1}^{K_i} w_{i,k} = 1$, with $\mathbf{W} = [w_{1,1}, \ldots, w_{c,K_c}]^\top$. The cumulative term $\sum_{j=1}^{K_i} w_{i,j}$ thus characterizes the contribution intensity of the i-th learnware toward fulfilling the requirement.

After deriving the relevance, mimic datasets are constructed from the identified learnwares to facilitate the training of a specification selector without any prior knowledge regarding the user task. Previous approaches [Chen *et al.*, 2025b] generate mimic training data by sampling from a multivariate Gaussian distribution, whose parameters are estimated from the mean and covariance of specifications aggregated at the task level. Such a coarse-grained generation strategy neglects the inherent multi-modal structure of real data, yielding mimic samples that fail to capture meaningful local variations and consequently weakening the selector's generalization performance.

The distinctive specifications generated during the submission stage help mitigate this challenge, as they lead to increased diversity in the mixture data generated from them. Building on this, we further explore the class-level data structure and local statistical properties to enhance sample diversity and discriminability for training the selector, thereby enabling more accurate model selection for each instance.

First, each specification $\Phi_{i,k}$ is divided into clusters, with the cluster count automatically determined by the silhouette coefficient according to Eq.(10).

$$S = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{10}$$

$$a(i) = \frac{1}{|C_i| - 1} \sum_{\substack{\boldsymbol{z}_j \in C_i \\ j \neq i}} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|, \tag{11}$$

$$b(i) = \min_{C \neq C_i} \frac{1}{|C|} \sum_{\boldsymbol{z}_j \in C} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|. \tag{12}$$

Here, $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are examples within the specification $\Phi_{i,k}$, and C_i denotes the class label of $\boldsymbol{z}_i$. Leveraging this approach, we can adaptively discern the local structural characteristics of these specifications, effectively integrating them into subsequent example generation.

For each obtained cluster C_i, the local statistical features are represented by the mean vector $mean(\boldsymbol{z}_i)$ and covariance matrix Σ_i of the contained samples. Based on these statistics, mimic data can be sampled according to (13).

$$\left\{X_i^{\text{mimic}}\right\}_{n=1}^{N_i^{\text{mimic}}} \sim \mathcal{N}\left(\text{mean}\left(\boldsymbol{z}_i\right), \rho(S_i) \cdot \boldsymbol{\Sigma}_i\right), \tag{13}$$

where $N_{i,k}^{\text{mimic}} = \hat{N} \cdot \sum_{k=1}^{K_i} \cdot w_{i,k}/|C|$, and $\rho(S_i) = exp(1 - S_i)$. This strategy adaptively modulates the generation behavior according to the clarity of cluster structures: when clusters are well-defined, the generated samples concentrate around cluster centers; conversely, in regions with ambiguous cluster boundaries, the generation becomes more dispersed to account for structural uncertainty. Once the mimic data are generated, they are combined with their corresponding specification indices to form a training set for learning a learnware selector $g(\cdot)$, enabling each instance within the user's task to be assigned to the most relevant learnware.

The utilization of the above class-level structural patterns and local statistical properties within obtained distinctive specifications overcomes the limitations of a single Gaussian distribution in preserving structural integrity and representing complex data distributions [Chen *et al.*, 2025b], thereby enabling the generated mimic data to maintain overall distributional consistency while substantially enhancing inter-sample diversity. As a result, the learned learnware index selector exhibits stronger generalization and enables more precise model selection. The detailed procedure of the entire deploying stage is provided in Algorithm 2 of Appendix C.

5 Experiments

5.1 Experimental Setup

Datasets. To validate the learnware paradigm, we derive datasets from *CIFAR100* [Krizhevsky, 2009] and *TinyImageNet200* [Le and Yang, 2015] and reconstruct them accordingly. *CIFAR100* consists of 100 classes that can be organized into 20 superclasses, while *TinyImageNet200* contains 200 classes that can be grouped into 40 superclasses. This superclass hierarchy fits well with the learnware paradigm, as each superclass corresponds to a distinct task. In the submitting stage, models trained on the superclass-defined tasks, together with their specifications, are packaged into learnwares and submitted to the learnware dock system. During the developing stage, the system then retrieves one or more learnwares whose specifications best align with the user's task.

Implementation Details. To ensure a fair comparison and eliminate variability arising from different feature extractors, all learnware-based baselines adopt the same *ResNet110*[1] backbone for image representation. Moreover, in submitting

[1]ResNet110 is trained by using the command provided at https://github.com/bearpaw/pytorch-classification/blob/master/TRAINING.md

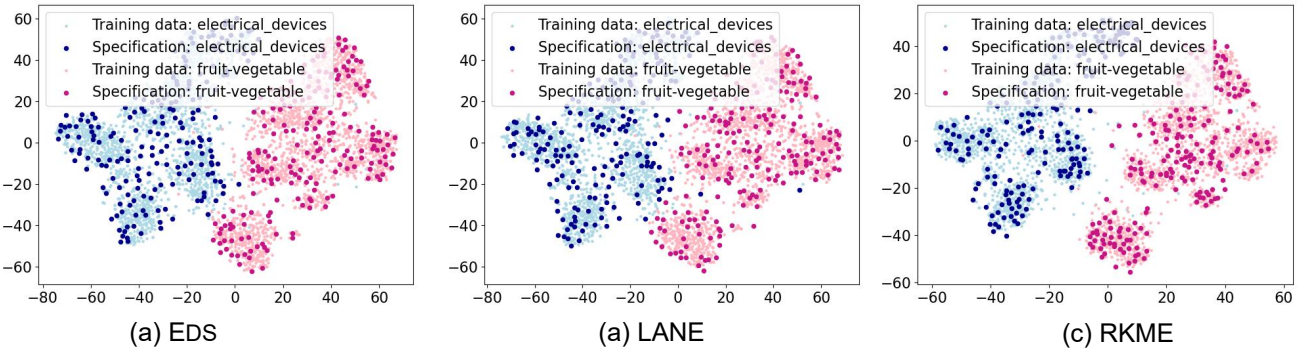

Figure 2: Visualization of training data and their corresponding specifications from a task-oriented perspective

stage, the stochastic network ψ_ϑ used in LANE [Chen *et al.*, 2025b] and the proposed EDS is implemented using the *ConvNetBN* [Rawat and Wang, 2017] architecture. In the deploying stage, the learnware index selector is realized via an SVM, which assigns the most relevant learnware to each requirement instance. More implementation details are illustrated in Appendix D.

Mixed tasks	1	2	5	10	20
MAX	67.64	68.56	67.86	67.38	67.61
HRM	68.91	68.15	68.28	67.98	67.75
RKME	84.04	81.78	78.17	66.11	66.29
LANE	84.53	83.65	77.87	72.21	67.63
EDS	**85.06**	**83.94**	**78.18**	**72.87**	**68.94**

Table 1: Results of CIFAR100 in Accuracy(%).

Mixed tasks	1	2	3	4	5
MAX	64.74	64.54	66.03	66.08	60.55
HMR	65.05	64.85	66.37	66.09	60.77
RKME	87.88	73.82	68.23	62.19	59.61
LANE	**90.48**	76.16	69.92	65.07	61.02
EDS	**90.48**	**76.80**	**70.47**	**66.20**	**61.57**

Table 2: Results of TinyImageNet200 in Accuracy(%).

5.2 Comparison to State-of-The-Art Methods

To demonstrate the effectiveness and superiority of the proposed EDS method, we conduct comparative experiments against four approaches: a naive baseline MAX, HMR [Wu *et al.*, 2019], RKME [Wu *et al.*, 2023], and LANE [Chen *et al.*, 2025b]. More comparative experiment details are provided in the Appendix D. Tables 1 and 2 present the model reuse accuracy of different algorithms on CIFAR100 and TinyImageNet200, respectively. The results demonstrate that the proposed EDS consistently achieves superior effectiveness and performance across both single-task and mixed-task settings.

To more intuitively highlight the superiority of the EDS method in specification quality, we employ t-SNE [Maaten and Hinton, 2008] to visualize training data from different domains along with their corresponding specifications generated by various methods. From Figure 2, although the central blue region contains abundant data, LANE and RKME fail to generate corresponding specifications, whereas EDS successfully does so, yielding specifications that more accurately characterize the tasks in which the model excels. Besides, while LANE allows specifications to capture the task feature distributions that models master, it lacks discriminability across tasks, resulting in blurred boundaries. RKME generates highly discriminative specifications, but at the cost of distributional alignment, leaving boundary regions with abundant training data without coverage. In contrast, EDS simultaneously preserves distributional alignment and explicitly enhances specification distinctiveness, enabling more precise model identification and effective reuse.

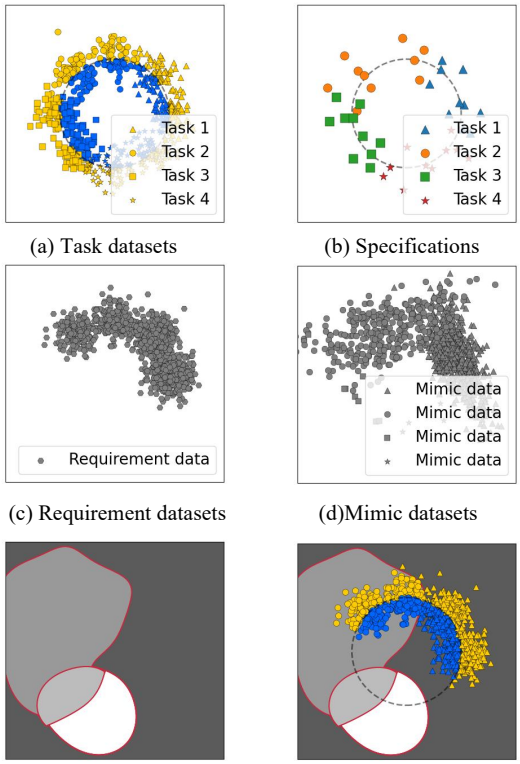

Figure 3: Visualization of synthetic samples generated under the learnware paradigm using EDS.

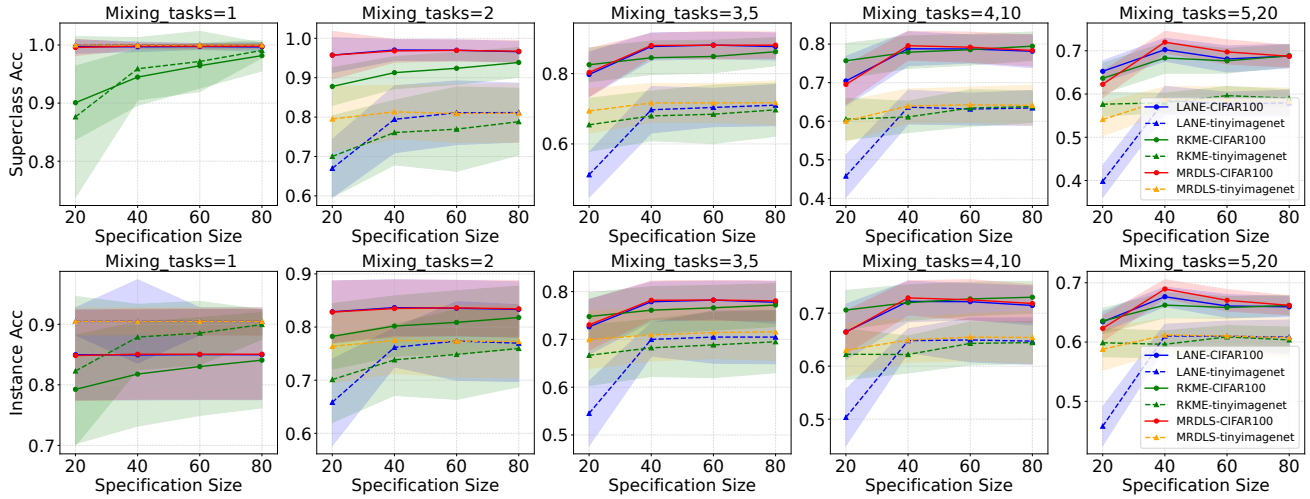

Figure 4: Effect of specification size on accuracy at both superclass and instance levels. The vertical axis denotes accuracy, while the horizontal axis represents the specification size.

5.3 Further Analyses

Visualization and Validation. For a more intuitive illustration of the effectiveness of the proposed method in the learnware paradigm, experiments are conducted on a synthetic dataset with visualized results. Initially, datasets corresponding to binary classification tasks are generated from four synthetic datasets, as shown in Figure 2(a). The proposed method is then applied to generate specifications for the four datasets, with a specification size of five per class. As illustrated in Figure 2(b), clear discriminability among different specifications can be observed. During the deploying stage, two closely related datasets are combined to simulate user requirements, as illustrated in Figure 2(c). When applied to these simulated requirements, EDS is able to accurately estimate the contribution of each dataset, recovering the true mixture weights $(0.7, 0.3, 0.0, 0.0)$ as $(0.699, 0.298, 0.002, 0.001)$, highlighting its precision in identifying the most relevant learnwares. The corresponding mimic data are then generated according to the estimated mixture weights, as shown in Figure 2(d), exhibiting a close resemblance to the user requirement data. Finally, the learnware index selector is trained on the synthetic mimic data (Figure 2(e)), with color blocks indicating predicted learnware indices. Applied to the user requirement instances, it attains a prediction accuracy of 98.8% (Figure 2(f)). These findings validate the effectiveness of EDS in the learnware paradigm.

Ablation Studies. The two primary contributions of this work are the submitting and deploying stages based on EDS, hereafter denoted as EDS-S and EDS-D, respectively. To evaluate the effectiveness of these two components, ablation studies are conducted, and the corresponding results are reported in Table 3. The experimental results demonstrate that both modules contribute positively to the final outcome, as incorporating either one individually leads to a steady improvement in model reuse, while the best performance is achieved when both modules are jointly employed.

Sensitivity Analysis The specification size is a critical factor in the learnware paradigm. To investigate its impact on model reuse performance, experiments under varying specification sizes are conducted, and the results are presented in Figure 4. It can be observed that EDS achieves superior performance over RKME and LANE under most specification sizes. The performance gain increases with the specification size and stabilizes thereafter, which can be attributed to the joint effects of specification generation in the submitting stage and mimic data construction in the deploying stage. Specifically, as suggested by Eq. (5) and Eq. (13), larger specifications retain more informative statistics of the original data and better approximate intra-class distributions, enabling the generation of more diverse mimic samples that align well with task data. Nevertheless, when the data distribution is already well captured, further enlarging the specification size brings limited additional benefits. δ is also an important parameter, and a sensitivity analysis of δ is provided in Appendix E.

EDS-S	EDS-D	Accuracy
–	✓	82.04
✓	–	83.53
✓	✓	**85.06**

Table 3: Ablation study on two modules of EDS. "✓" denotes the model is included, and "–" otherwise.

6 Conclusion

In this paper, we propose an ***E**xplicitly* **D***istinctive* **S***pecification* (EDS) method, which explicitly incorporates inter-specification discriminability at the submitting stage for the first time. Building on this, class-level data structure and local statistical properties among these specifications are further explored to preserve distributional consistency while increasing sample diversity for selector training in the deploying stage. Extensive experiments demonstrate that the proposed EDS method can advance a more discriminative learnware paradigm, enabling more accurate and efficient learnware identification and reuse.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Science Foundation of China under Grant NSF 62422604.

References

[Anaby-Tavor *et al.*, 2020] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. Do not have enough data? deep learning to the rescue! In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pages 7383–7390. AAAI Press, 2020.

[Chen *et al.*, 2025a] Wei Chen, Jun-Xiang Mao, Xiaozheng Wang, and Min-Ling Zhang. Learnware specification via dual alignment. In *Forty-second International Conference on Machine Learning*, 2025.

[Chen *et al.*, 2025b] Wei Chen, Jun-Xiang Mao, and Min-Ling Zhang. Learnware specification via label-aware neural embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 15857–15865, 2025.

[Dong *et al.*, 2022] Tian Dong, Bo Zhao, and Lingjuan Lyu. Privacy for free: How does dataset condensation help privacy? In *International Conference on Machine Learning*, pages 5378–5396. PMLR, 2022.

[Guo *et al.*, 2023] Lan-Zhe Guo, Zhi Zhou, Yu-Feng Li, and Zhi-Hua Zhou. Identifying useful learnwares for heterogeneous label spaces. In *Proceedings of the 40th International Conference on Machine Learning*, pages 12122–12131, Honolulu, Hawaii, 2023.

[Jordan and Mitchell, 2015] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[Krizhevsky, 2009] Alex Krizhevsky. Poligon: A system for parallel problem solving. Technical Report Online, Available: https://www.cs.toronto.edu/ kriz/cifar.html, 2009. Accessed: 2025-01-06.

[Le and Yang, 2015] Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[Lei *et al.*, 2024] Hao-Yi Lei, Zhi-Hao Tan, and Zhi-Hua Zhou. On the ability of developers' training data preservation of learnware. In *Advances in Neural Information Processing Systems 37*, Vancouver, Canada, 2024.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[Mohri and Muñoz Medina, 2012] Mehryar Mohri and Andres Muñoz Medina. New analysis and algorithm for learning with drifting distributions. In *International Conference on Algorithmic Learning Theory*, pages 124–138. Springer, 2012.

[Rawat and Wang, 2017] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[Shazeer and et al., 2018] Noam Shazeer and et al. Mesh-tensorflow: Deep learning for supercomputers. In *Advances in Neural Information Processing Systems (NeurIPS) 2018*, 2018.

[Shejwalkar and Houmansadr, 2021] Virat Shejwalkar and Amir Houmansadr. Membership privacy for machine learning models through knowledge transfer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 9549–9557, 2021.

[Tan *et al.*, 2023] Peng Tan, Zhi-Hao Tan, Yuan Jiang, and Zhi-Hua Zhou. Handling learnwares developed from heterogeneous feature spaces without auxiliary data. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, pages 4235–4243, Macao, China, 2023.

[Tan *et al.*, 2024a] Peng Tan, Hai-Tian , Zhi-Hao Tan, and Zhi-Hua Zhou. Handling learnwares from heterogeneous feature spaces with explicit label exploitation. In *Advances in Neural Information Processing Systems 37*, Vancouver, Canada, 2024.

[Tan *et al.*, 2024b] Peng Tan, Zhi-Hao Tan, Yuan Jiang, and Zhi-Hua Zhou. Towards enabling learnware to handle heterogeneous feature spaces. *Machine Learning*, 113(3):1839–1860, 2024.

[Tan *et al.*, 2024c] Zhi-Hao Tan, Jian-Dong Liu, Xiao-Dong Bi, Peng Tan, Qin-Cheng Zheng, Hai-Tian Liu, Yi Xie, Xiao-Chuan Zou, Yang Yu, and Zhi-Hua Zhou. Beimingwu: A learnware dock system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5773–5782, Barcelona, Spain, 2024.

[von Rueden *et al.*, 2019] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Michal Walczak, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. Informed machine learning–a taxonomy and survey of integrating knowledge into learning systems. *arXiv preprint arXiv:1903.12394*, 2019. Accepted for publication in IEEE Transactions on Knowledge and Data Engineering.

[Wu *et al.*, 2019] Xi-Zhu Wu, Song Liu, and Zhi-Hua Zhou. Heterogeneous model reuse via optimizing multiparty multiclass margin. In *International Conference on Machine Learning*, pages 6840–6849. PMLR, 2019.

[Wu *et al.*, 2023] Xi-Zhu Wu, Wenkai Xu, Song Liu, and Zhi-Hua Zhou. Model reuse with reduced kernel mean embedding specification. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):699–710, 2023.

[Xie *et al.*, 2023] Yi Xie, Zhi-Hao Tan, Yuan Jiang, and Zhi-Hua Zhou. Identifying helpful learnwares without exam-

ining the whole market. In *Proceedings of the 26th European Conference on Artificial Intelligence*, pages 2752–2759, Kraków, Poland, 2023.

[Ye *et al.*, 2021] Han-Jia Ye, De-Chuan Zhan, Yuan Jiang, and Zhi-Hua Zhou. Heterogeneous few-shot model rectification with semantic mapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3878–3891, 2021.

[Zhang *et al.*, 2021] Yu-Jie Zhang, Yu-Hu Yan, Peng Zhao, and Zhi-Hua Zhou. Towards enabling learnware to handle unseen jobs. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 10964–10972, Virtual Event, 2021.

[Zhao and Bilen, 2021] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021.

[Zhou and Tan, 2024] Zhi-Hua Zhou and Zhi-Hao Tan. Learnware: Small models do big. *Science China Information Sciences*, 67(1):112102, 2024.

[Zhou, 2016] Zhi-Hua Zhou. Learnware: on the future of machine learning. *Frontiers of Computer Science*, 10(4):589–590, 2016.