A Unifying Perspective on Model Reuse: From Small to Large Pre-Trained Models

Da-Wei Zhou, Han-Jia Ye(✉)
School of Artificial Intelligence, Nanjing University
National Key Laboratory for Novel Software Technology, Nanjing University
{zhoudw, yehj}@lamda.nju.edu.cn

Abstract

Machine learning has rapidly progressed, resulting in a vast repository of both general and specialized models that address diverse practical needs. Reusing pre-trained models (PTMs) from public model zoos has emerged as an effective strategy, leveraging rich model resources and reshaping traditional machine learning workflows. These PTMs encapsulate valuable inductive biases beneficial for downstream tasks. Well-designed reuse strategies enable models to be adapted beyond their original scope, enhancing both performance and efficiency in target machine learning systems. This survey offers a unifying perspective on model reuse, establishing connections across various domains and presenting a novel taxonomy that encompasses the full lifecycle of PTM utilization—including selection from model zoos, adaptation techniques, and related areas such as model representation learning. We delve into the similarities and distinctions between reusing specialized and general PTMs, providing insights into their respective advantages and limitations. Furthermore, we discuss key challenges, emerging trends, and future directions in model reuse, aiming to guide research and practice in the era of large-scale pre-trained models. A comprehensive list of papers about model reuse is available at https://github.com/LAMDA-Model-Reuse/Awesome-Model-Reuse.

1 Introduction

Recent years have witnessed the rapid advancement of machine learning, achieving competitive performance across various real-world applications. A typical machine learning pipeline consists of two core steps: collecting data and training a model [Mitchell, 1997], as illustrated in Figure 1 (top). This *data-centric* paradigm heavily relies on the quantity and quality of training data to ensure strong generalization. However, in resource-constrained settings, acquiring high-quality data at scale poses significant challenges, limiting the effectiveness of traditional machine learning approaches.

To this end, researchers are exploring how to reduce the cost by *reusing existing knowledge*. In this way, reusing Pre-Trained Models (PTMs) rather than training models from scratch—has emerged as a promising solution for knowledge transfer, especially with the growing availability of PTMs across diverse domains [Wolf, 2019]. This paradigm leverages a vast array of PTMs that encapsulate valuable inductive biases, making them highly beneficial for various downstream tasks. With well-designed reuse strategies, knowledge from these PTMs can be efficiently extracted and extended beyond their original scope, facilitating numerous applications [Zhou, 2016; Zhou and Tan, 2024; Tan *et al.*, 2025; Lei *et al.*, 2024].

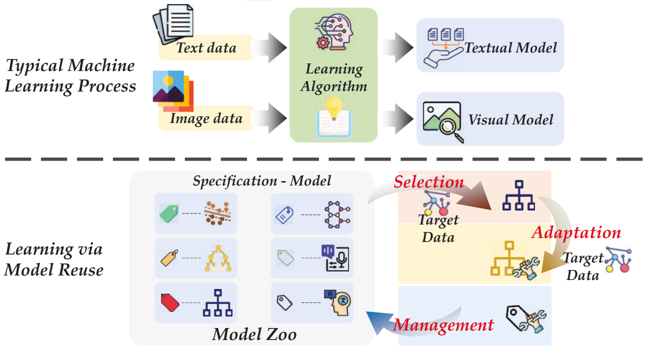

Figure 1: Comparison between typical machine learning (top) and learning via model reuse (bottom). In the typical paradigm, machine learning relies on collecting task-specific data and training models from scratch. In contrast, the model reuse paradigm leverages a repository of Pre-Trained Models (PTMs), known as a model zoo, where models are selected and adapted for new tasks. The model zoo also supports the management and maintenance of these reusable PTMs. Models within the zoo are organized in a key-value structure, where each model is indexed by a key (*e.g.*, a specification).

The idea of reusing PTMs dates back to the 1990s when researchers explored the use of pre-trained neural networks as feature extractors for tasks such as waveform and handwritten character recognition [Guo and Gelfand, 1992]. As PTMs have grown more powerful—especially those trained on large-scale datasets—model reuse has emerged as a resource-efficient and performance-competitive alternative to training models from scratch. The concept of model reuse was formally introduced in [Zhou, 2016], highlighting the *reusable* property of machine learning models. Early implementations of model reuse include vanilla fine-tuning [Yosinski *et al.*, 2014] and biased regularization [Kuzborskij and Orabona, 2017]. However, to address task-specific challenges such as distribution shifts and sample scarcity [Zheng *et al.*, 2023; Zhuang *et al.*, 2020], more advanced model reuse techniques have been developed across various areas [Kundu *et al.*, 2020; Hinton *et al.*, 2015; Chen *et al.*, 2022].

With the rapid expansion of machine learning, an increasing number of PTMs are being developed and organized into public PTM repositories [Wolf, 2019; Tan *et al.*, 2024], enabling the next evolution of model reuse based on a model zoo. The diversity of PTMs within a model zoo—ranging from small specialized PTMs tailored for domains such as medicine, finance, and education [Luo *et al.*, 2024], to large foundation models with strong zero-shot capabilities [Bommasani *et al.*, 2021]—introduces new challenges and opportunities. As shown in Figure 1 (bottom), a more advanced model reuse paradigm with a model zoo involves several key steps [Zhou, 2016; Zhou and Tan, 2024], *e.g.*, (1) selecting appropriate PTMs from the model zoo, (2) leveraging the selected PTMs to help the learning of the target task, and (3) managing and improving the model zoo for easier selection and enhanced overall performance.

Despite the success of model reuse, two key challenges remain in comprehensively exploring its core principles. First, as model reuse becomes a natural choice in various applications, a holistic and task-agnostic perspective is needed to unify fragmented research efforts across different fields. Establishing a big-picture view of model reuse can bridge the gaps between specialized subfields and promote broader applicability. Second, with the emergence of large PTMs such as foundation models [Bommasani *et al.*, 2021], reusing such models involves both shared principles and diverse implementations. Integrating model reuse across both small and large PTMs can help connect diverse methodologies and inspire the design of novel model reuse strategies.

To tackle these existing challenges, this survey provides a unifying perspective on model reuse by categorizing research efforts based on two fundamental steps: *model selection and adaptation*. We introduce a novel taxonomy to organize existing approaches. Model selection methods are categorized based on the selection mechanism, namely, semantic/rule-based, metric-based, and learning-based strategies. Model adaptation methods are classified according to the role of PTMs in the learning process, including leveraging PTMs for data preparation, model training, and inference.

Additionally, we distinguish between small and large PTMs, recognizing that while there is no strict threshold, we define small PTMs as relatively compact, *specialized* models, whereas large PTMs refer to *general-purpose* models trained on diverse data with strong *zero-shot* generalization across domains. By analyzing both their commonalities and differences, we highlight the advantages, challenges, and broader impact of model reuse across the machine learning pipeline.

The main contributions of this survey include:

- A unifying framework for model reuse, categorizing research based on PTM selection and adaptation strategies.
- A comprehensive review covering both small and large PTMs, highlighting their commonalities and differences.
- We summarize insights into emerging trends and challenges in the future of model reuse.

2 Preliminaries

2.1 Reusing PTMs from a Model Zoo

We illustrate the model reuse process using classification as an example, though the methodology can be extended to other tasks with different types of PTMs. In standard machine learning, a C-class classification task consists of a training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ with N examples, where each instance $\boldsymbol{x}_i \in \mathbb{R}^d$ and its corresponding label $y_i \in [C] = \{1, \ldots, C\}$. The goal is to construct a classifier $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^C$ with parameters $\boldsymbol{\theta}$ that maps an input instance to C-dimensional confidence scores. Typically, f represents a neural network, and $\boldsymbol{\theta}$ corresponds to its weights, which are learned through empirical risk minimization.

Reusing a Single PTM. Instead of training $f_{\boldsymbol{\theta}}$ from scratch, a well-trained model $g_{\boldsymbol{\Theta}}$ is often available and can be leveraged to facilitate the training of $f_{\boldsymbol{\theta}}$. Here, $g_{\boldsymbol{\Theta}}$ is pre-trained on a dataset $\mathcal{D}' = \{(\boldsymbol{x}'_j, y'_j)\}_{j=1}^{N'}$ with instances $\boldsymbol{x}'_j \in \mathbb{R}^{d'}$ and labels $y'_j \in [C']$. To reuse the expert knowledge in $g_{\boldsymbol{\Theta}}$, an adaptation strategy is applied $f_{\boldsymbol{\theta}} = \textbf{Adapt}(f_{\boldsymbol{\theta}_0} \mid \mathcal{D}, g_{\boldsymbol{\Theta}})$, where $\boldsymbol{\theta}_0$ is the initial state of the model before adaptation. Since the PTM $g_{\boldsymbol{\Theta}}$ may differ from $f_{\boldsymbol{\theta}}$ in various aspects—such as model architecture ($f \neq g$), data distribution ($\Pr(\mathcal{D}) \neq \Pr(\mathcal{D}')$), feature dimension ($d \neq d'$), or label space ($C \neq C'$)—the adaptation function $\textbf{Adapt}(\cdot)$ must address these heterogeneities.

Reusing Multiple PTMs from a Model Zoo. With the proliferation of PTMs across different domains, model zoos, denoted as $\mathcal{M} = \{g_{\boldsymbol{\Theta}}^1, \ldots, g_{\boldsymbol{\Theta}}^M\}$, have emerged as valuable resources. These PTMs encode diverse inductive biases, making them highly beneficial for a wide range of target tasks. To effectively utilize these rich model repositories, [Zhou, 2016; Zhou and Tan, 2024] summarizes and highlights the importance of a "select-then-adapt" workflow. For simplicity, consider selecting a single PTM, and scenarios selecting multiple PTMs could be easily extended. In the *model selection* stage, a selection mechanism ranks the PTMs in $\mathcal{M}$ based on their relevance to the target task $\mathcal{D}$ and outputs the index of the most appropriate PTM:

$$m = \arg\max\nolimits_{m \in [M]} \textbf{Select}(\mathcal{D}, f_{\boldsymbol{\theta}} \mid \mathcal{M}) \; . \tag{1}$$

$f_{\boldsymbol{\theta}}$ in Eq. 1 indicates the selection process also depends on the required architecture f over the target task. The workflow steps into the *model adaptation* stage once a PTM is selected, which is applied as follows:

$$f_\theta = \textbf{Adapt}(f_{\boldsymbol{\theta}_0} \mid \mathcal{D}, g_{\boldsymbol{\Theta}}^m) \; . \tag{2}$$

Model selection ensures the most relevant PTM is identified, while adaptation efficiently bridges the gap between the PTM and the target task. The adaptation could be applied to the selected model itself (*i.e.*, adapt $g_{\boldsymbol{\Theta}}^m$) or to the target model $f_{\boldsymbol{\theta}}$ where the selected $g_{\boldsymbol{\Theta}}^m$ provides auxiliary expert knowledge.

2.2 Small and Large PTMs

Model zoos often contain PTMs of varying types, including those pre-trained on diverse domains such as finance, education, and scientific research. These PTMs also differ in functionality, covering tasks such as classification, regression, object detection, image/text recognition, text generation, and reinforcement learning.

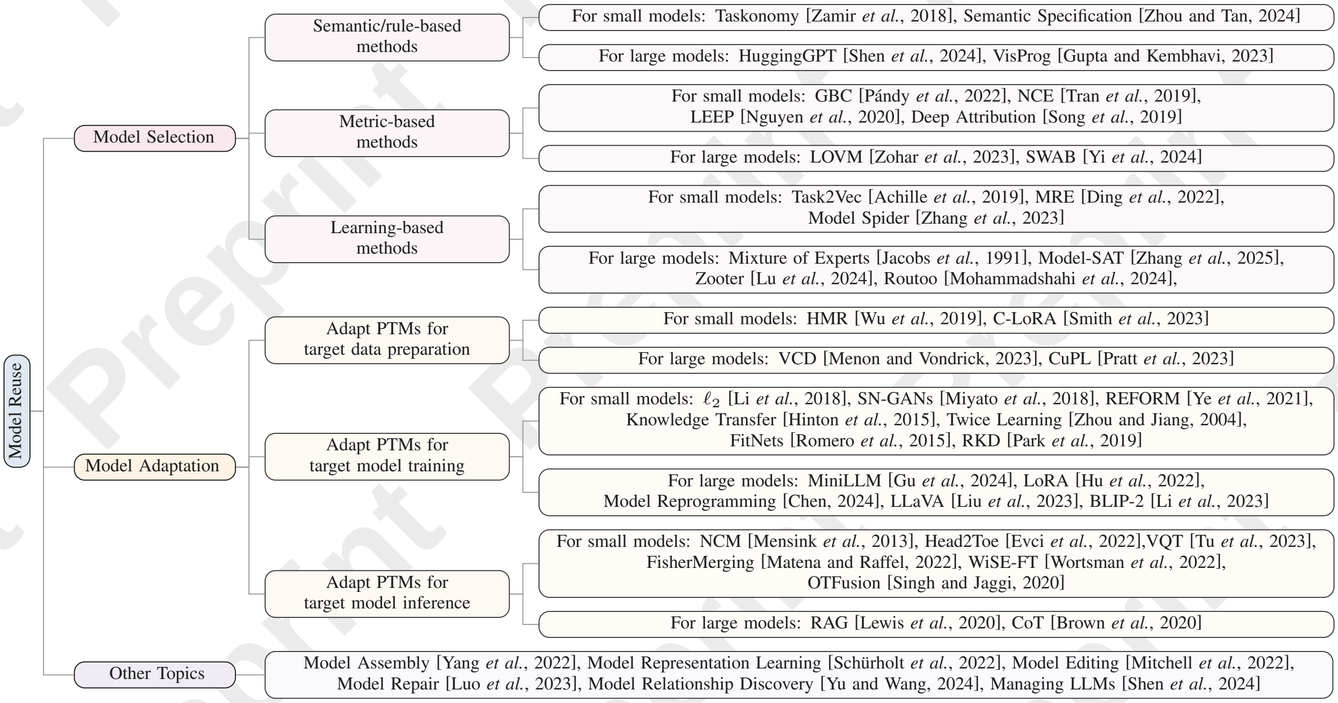
Model Reuse
Model Selection
Semantic/rule-based methods
For small models: Taskonomy [Zamir et al., 2018], Semantic Specification [Zhou and Tan, 2024]
For large models: HuggingGPT [Shen et al., 2024], VisProg [Gupta and Kembhavi, 2023]
Metric-based methods
For small models: GBC [Pándy et al., 2022], NCE [Tran et al., 2019], LEEP [Nguyen et al., 2020], Deep Attribution [Song et al., 2019]
For large models: LOVM [Zohar et al., 2023], SWAB [Yi et al., 2024]
Learning-based methods
For small models: Task2Vec [Achille et al., 2019], MRE [Ding et al., 2022], Model Spider [Zhang et al., 2023]
For large models: Mixture of Experts [Jacobs et al., 1991], Model-SAT [Zhang et al., 2025], Zooter [Lu et al., 2024], Routoo [Mohammadshahi et al., 2024],
Model Adaptation
Adapt PTMs for target data preparation
For small models: HMR [Wu et al., 2019], C-LoRA [Smith et al., 2023]
For large models: VCD [Menon and Vondrick, 2023], CuPL [Pratt et al., 2023]
Adapt PTMs for target model training
For small models: ℓ_2 [Li et al., 2018], SN-GANs [Miyato et al., 2018], REFORM [Ye et al., 2021], Knowledge Transfer [Hinton et al., 2015], Twice Learning [Zhou and Jiang, 2004], FitNets [Romero et al., 2015], RKD [Park et al., 2019]
For large models: MiniLLM [Gu et al., 2024], LoRA [Hu et al., 2022], Model Reprogramming [Chen, 2024], LLaVA [Liu et al., 2023], BLIP-2 [Li et al., 2023]
Adapt PTMs for target model inference
For small models: NCM [Mensink et al., 2013], Head2Toe [Evci et al., 2022],VQT [Tu et al., 2023], FisherMerging [Matena and Raffel, 2022], WiSE-FT [Wortsman et al., 2022], OTFusion [Singh and Jaggi, 2020]
For large models: RAG [Lewis et al., 2020], CoT [Brown et al., 2020]
Other Topics
Model Assembly [Yang et al., 2022], Model Representation Learning [Schürholt et al., 2022], Model Editing [Mitchell et al., 2022], Model Repair [Luo et al., 2023], Model Relationship Discovery [Yu and Wang, 2024], Managing LLMs [Shen et al., 2024]

Figure 2: A taxonomy of model reuse.

The emergence of large PTMs has significantly impacted the model reuse workflow, as computational costs have become a crucial factor, and there are diverse strategies for extracting expert knowledge from these models. While reusing both small and large PTMs shares fundamental principles, the differences in scale introduce distinct challenges and opportunities. This survey connects the reuse of both types of PTMs, highlighting their commonalities and differences to provide deeper insights into model reuse methodologies.

Since PTM size is relative, there is no strict boundary between small and large PTMs. Generally, we define small PTMs as models based on classical architectures and pre-trained for specialized tasks, such as ResNet for image classification [He *et al.*, 2016]. In contrast, large PTMs typically contain billions of parameters, are general-purpose, and possess zero-shot capabilities across various tasks. For example, a large language model (LLM) can handle text generation, classification, and reasoning without explicit task-specific fine-tuning, adapting dynamically to new queries [Zhao *et al.*, 2023].

2.3 Advantages and Goals

Model reuse facilitates *knowledge transfer* by leveraging pretrained models to improve generalization and reduce dependence on large labeled datasets. It also enables *knowledge aggregation*, where combining multiple PTMs enhances robustness, accuracy, and adaptability. By building on existing models, reuse accelerates training, lowers computational and data costs, and mitigates catastrophic forgetting in dynamic or continually evolving environments. Overall, model reuse improves scalability and efficiency while extending the applicability of machine learning across diverse tasks and domains.

3 Taxonomy

Model reuse consists of two core steps: model selection and model adaptation. Accordingly, we systematically categorize existing approaches into these two components, as illustrated in Figure 2. The objective of model selection is to identify relevant PTMs from the model zoo, which hinges on defining an appropriate retrieval metric. Based on how this matching is determined, we classify model selection methods into three categories: (i) semantic/rule-based methods, (ii) metric-based methods, and (iii) learning-based methods. These approaches differ in how they quantify model-task relevance. Once a PTM is selected, model adaptation ensures its effective integration into the target task. We categorize adaptation methods based on their application stage: (i) PTMs for data preparation, (ii) PTMs for model training, and (iii) PTMs for model inference. Since model reuse is a broad topic within machine learning, some related areas extend beyond the selection-adaptation paradigm. Therefore, we discuss model collaboration, compression, and representation learning, which contribute to the broader landscape of model reuse.

3.1 Model Selection

The target of model selection is to choose one or multiple related models from the model zoo that are suitable for the related dataset. This is a typical retrieval process, where the core problem is to *rank* all models in the model zoo correctly. Hence, we need to design a *metric* that measures the matching degree between models and the downstream tasks. For this perspective, we divide current approaches into three categories, each highlighting a different perspective to calculate the matching degree, *i.e.*, semantic/rule-based methods,

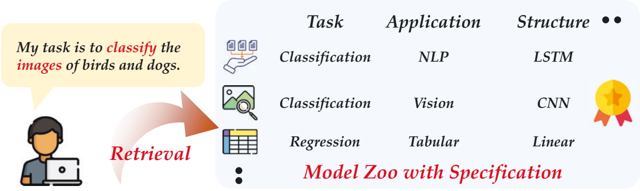

Figure 3: The pipeline of semantic/rule-based model selection. Models are organized with specifications, and users provide their requests regarding the dataset to retrieve the related model.

metric-based methods, and learning-based methods. Specifically, semantic/rule-based methods are designed to utilize the semantic information (*e.g.*, natural language descriptions) or manually designed rules to find related models. Metric-based methods rely on defining the matching degree between the model's output and the corresponding dataset, while learning-based methods directly learn the similarity between data and models. This topic is also regarded as measuring the transferability score of a PTM [Xue *et al.*, 2024].

Semantic/rule-based methods

For small models. Before the prosperity of large models, typical works rely on semantic information to choose proper models from the model zoo. This semantic information is commonly represented by natural languages, expressing the capability of models and the upstream training data. For example, [Zhou and Tan, 2024] proposes a typical model selection paradigm, where models in the model zoo are accompanied by descriptions (known as "semantic specification"). The semantic specification contains descriptive information related to the model, including the target task (*e.g.*, classification or regression), the kind of machine learning applications (*e.g.*, natural language processing or image recognition), the model structure (*e.g.*, SVM, CNN, or Decision Tree), etc. Hence, we can form a key containing all this related information as $\mathbf{s} = \{s_1, s_2, \cdots, s_k\}$, where s_i represents a dimension to describe the model. Hence, facing a new task, the users are only required to describe their requirements as a new query and search within the model zoo. Similar ideas are also explored in Taskonomy [Zamir *et al.*, 2018], where neural networks are adopted to find transfer learning dependencies among a wide array of tasks, leading to a structured approach that reduces the need for labeled data in training new tasks.

For large Models. Semantic specifications are manually described by the model owners, costing extra effort and domain knowledge. To this end, the prosperity of large models helps get rid of the manual description. HuggingGPT [Shen *et al.*, 2024] utilizes GPT to manage millions of models on HuggingFace [Wolf, 2019]. Similarly, natural language is utilized as the interface to align all models. Facing the user's new requirement, it plans out the necessary sub-tasks, and selects proper models from HuggingFace. Similar ideas have also been explored in automatically selecting and combining computer vision models [Gupta and Kembhavi, 2023].

Pros & Cons: We visualize the pipeline of semantic/rule-based methods in Figure 3. These methods provide a naive idea to manage models in the model zoo since natural language is a common tool for description. They can easily adapt to various domains and tasks by simply tagging models with appropriate semantic labels. This flexibility makes them suitable for dynamic environments where new types of tasks frequently emerge. However, there are also some drawbacks. The effectiveness of these methods heavily relies on the accuracy and comprehensiveness of the semantic specifications provided. Inaccurate or vague descriptions can lead to poor model selection, affecting overall performance. Besides, manual semantic tagging becomes impractical as the number of models grows. This is particularly challenging in environments where new models are constantly added [Wolf, 2019].

Metric-based methods

For small models. Apart from manually designing the specification, there are also works aiming to define the metric to represent the fitness between models and datasets. Correspondingly, metric-based methods determine whether a PTM is suitable for the target task by evaluating various proxy fitness metrics. There are various ways to design the metric, *e.g.*, using feature statistics [Pándy *et al.*, 2022], joint distribution of the source and target data [Nguyen *et al.*, 2020], mutual information-based transferability [Tran *et al.*, 2019], and gradients with few updates [Song *et al.*, 2019].

Specifically, we can utilize the pre-trained model to encode the downstream task's data, and evaluate the separability of per-class instances considering inter-class and intra-class information [Pándy *et al.*, 2022]. Other works seek to jointly model the upstream and downstream data. LEEP [Nguyen *et al.*, 2020] applies the PTM to the downstream dataset to estimate the label distribution for each input. Then, it assesses how well the predictions from the model align with the actual distribution of ground truth labels. The metric is based on the alignment of the model's predictions with the target task's requirements. NCE [Tran *et al.*, 2019] uses the conditional entropy between label sequences of source and target tasks as a metric. This measure reflects the information required to predict labels of one task based on the knowledge of another, thereby providing an estimate of task transferability and hardness. The aforementioned works only require the simple forward pass of the related dataset. However, if the computational resources are sufficient and can support backpropagation, we can also utilize the matching degree of attribution maps to define the metric [Song *et al.*, 2019]. This line of work is very similar to transferability estimation, and we refer readers to [Xue *et al.*, 2024] for more details.

For large models. In the era of large models, the cost of evaluating transferability becomes a critical factor due to the high computational expense involved in even simple operations like the forward pass. To address these challenges, recent developments focus on reducing the computational demands of metric-based evaluations. To this end, recent work focuses on designing surrogate tasks to approximate the behaviors of larger models without needing to operate them directly, thus speeding up the evaluation process. For example, LOVM [Zohar *et al.*, 2023] defines the vision-language model selection paradigm based solely on textual descriptions of tasks, without needing access to specific datasets. SWAB [Yi *et al.*, 2024] enhances the selection performance by bridging the modality gap between the textual descriptions and the visual model.

Pros & Cons. Metric-based methods provide a quantitative measure of suitability, making comparisons between models straightforward and reducing subjective bias in model selection. Besides, they adapt well to different types of data and tasks by adjusting the metrics to reflect the specific needs of the target task. However, the effectiveness of metric-based methods heavily relies on the appropriateness and robustness of the chosen metric. Poorly designed metrics can lead to suboptimal model selection. Some metrics, especially those involving backpropagation or complex statistical analyses, can be computationally intensive, limiting their use in resource-constrained environments.

Learning-based methods

For small models. Apart from manually describing the models' behaviors, there are also works on automatically learning the specifications. This line of work aims to project models, as well as datasets, into the same embedding space, where the retrieval process can be easily done by distance calculation in the unified space. A representative work, *i.e.*, Task2Vec [Achille *et al.*, 2019] involves passing data through a probe network to generate a task-specific embedding that captures the complexity and characteristics of the task. Recent papers in this area also involve *learning the specifications* of PTMs and tasks from data. By training an encoder on known tasks and PTMs, these methods enable the generalization of the ability to represent both datasets and PTMs to unseen tasks and PTMs [Ding *et al.*, 2022]. Model Spider [Zhang *et al.*, 2023] learns model representations through vast historical performance.

For large models. Learning-based methods enable LLM selection with adaptive characteristics [Guha *et al.*, 2024; Muqeeth *et al.*, 2024]. A representative work is mixture of experts (MoE) [Jacobs *et al.*, 1991], where multiple experts are constructed for diverse tasks. The training stage will adaptively assign the weight of each expert for the aggregated prediction to identify the best fit for each piece of reused tasks. Recent advancements in LLM selection have focused on learning effective routers for multiple pre-trained LLMs [Pfeiffer *et al.*, 2020; Zhang *et al.*, 2023; Ong *et al.*, 2025; Wu *et al.*, 2024; Jitkrittum *et al.*, 2025; Frick *et al.*, 2025], *e.g.*, Zooter [Lu *et al.*, 2024] employs a reward model to score query-output pairs for routing decision-making, while [Mohammadshahi *et al.*, 2024] uses self-play in reinforcement learning to create query-response-score triplets. Model-SAT [Zhang *et al.*, 2025] further utilizes an extra foundation model as the model selector.

Pros & Cons. Learning-based methods automate the specification learning process, enabling scalability by adapting to new models and tasks in an end-to-end manner. However, their effectiveness heavily depends on the quality and diversity of the training data. Inadequate or biased data can lead to poor performance. Besides, the generalization ability of these works is also highly dependent on the scale of training tasks. Finally, the process of training and deploying these systems, especially for large models, can be resource-intensive, requiring substantial computational power and time.

3.2 Adapting PTMs

After selecting models from the model zoo, we then discuss how the selected model can help obtain better performance in the target task. In the following section, we will explore how PTMs can be effectively integrated into different learning phases, namely, target data preparation, target model training, and target model inference.

Adapt PTMs during target data preparation

For small models. PTMs serve as experts capable of generating enriched data for the target task, enabling the target model to achieve better generalization ability. For example, generative PTMs model the distribution of pretext tasks, and utilizing them to generate related datasets can help the training process of resource-constrained scenarios [Smith *et al.*, 2023]:

$$f_\theta = \textbf{Adapt}(f_{\boldsymbol{\theta}_0} \mid \mathcal{D} \cup \mathcal{D}^m_{\boldsymbol{\Theta}}, g^m_{\boldsymbol{\Theta}}) \; , \tag{3}$$

where $\mathcal{D}^m_{\boldsymbol{\Theta}}$ is generated by the selected model $g^m_{\boldsymbol{\Theta}}$. The enriched data can also emerge in other forms, *e.g.*, the dataset statistics can help calibrate the distribution of downstream tasks [Wu *et al.*, 2019]. The output of PTM can also be applied to tackle black-box source-free adaptation [Liang *et al.*, 2022] via knowledge adaptation.

For large models. The advancement of large language models offers a new insight into the zero-shot ability, which can be further utilized for data preparation. Some recent methods also treat the pre-trained LLMs as a knowledge base and retrieve the common knowledge in the text form [Menon and Vondrick, 2023; Pratt *et al.*, 2023]. In this case, the word-level labels are further refined into the semantic level (*e.g.*, "hen" → "two legs", "red, brown, or white feathers", "a small body", "a small head", etc.) to further facilitate recognition.

Pros & Cons. With the help of PTMs, we can improve the diversity and quality of datasets, which is crucial for training robust models. The enriched data help models learn more generalized features, potentially improving performance on unseen data. However, there are also some drawbacks. The effectiveness of the data preparation heavily relies on the quality of the PTMs used. If the PTMs are biased or trained on non-representative data, this can adversely affect the quality of the prepared data. Besides, there is a risk that models may overfit the characteristics of data generated or modified by PTMs, especially if the diversity of the synthetic data is limited or too closely mirrors the training scenarios of the PTMs.

Adapt PTMs during target model training

For small models. With the selected pre-trained model, an intuitive way to apply it to downstream tasks is fine-tuning. PTMs, when related to the target task, offer a strong foundation upon which the target model can be built:

$$f_\theta = \textbf{Adapt}(f_{\boldsymbol{\theta}_0} \mid \mathcal{D}, g^m_{\boldsymbol{\Theta}}) \quad \text{with} \quad f_{\boldsymbol{\theta}_0} = g^m_{\boldsymbol{\Theta}} \; . \tag{4}$$

In Eq. 4, the selected PTM is directly applied as the initialization of the target model. Fine-tuning these PTMs often leads to significant improvements in the target model's performance [Yosinski *et al.*, 2014].

Since PTMs have been extensively trained on the pretext task, directly fine-tuning will ruin the existing knowledge contained in the PTM. Hence, there are some works with

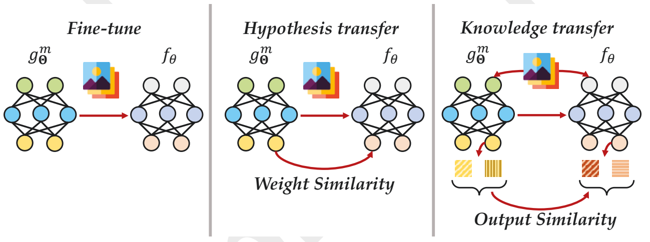

Figure 4: The illustrations of adapting PTMs during target model training. PTMs can serve as the initialization, providing weight regularization and output supervision to help target tasks.

regularization to regularize PTM adaptation. Hypothesis transfer serves as a regularization technique that encourages the weights of the target model to resemble those of the PTM.

$$\arg\min_{\theta} \mathcal{L}(f_{\boldsymbol{\theta}_0} \mid \mathcal{D}, g^m_{\boldsymbol{\Theta}}) + \lambda \mathcal{R}(f_\theta, g^m_{\boldsymbol{\Theta}}) \,, \tag{5}$$

where $\mathcal{R}(f_\theta, g^m_{\boldsymbol{\Theta}})$ denotes the regularization term on the similarity between $(f_\theta, g^m_{\boldsymbol{\Theta}})$. This is achieved by defining similarity measures in weight space, effectively guiding the target model to be close to the PTM. Various norm-based methods, such as the Frobenius Norm, ℓ_2-norm [Li *et al.*, 2018], and spectral norm [Miyato *et al.*, 2018] are employed to quantify the similarity between the weights. When dealing with weight spaces of diverse shapes, additional mapping and attention layers [Ye *et al.*, 2021] are applied to facilitate alignment.

Another line of work considers utilizing PTMs as the external supervision signal. Knowledge distillation was thought to be proposed by [Hinton *et al.*, 2015], while ten years earlier a similar idea has already been published by [Zhou and Jiang, 2004]. These approaches match the PTM and the target model in the prediction space, *i.e.*, the output predictions of both models should be similar:

$$\arg\min_{\theta} \mathcal{L}(f_{\boldsymbol{\theta}_0} \mid \mathcal{D}, g^m_{\boldsymbol{\Theta}}) - \lambda \Sigma_{\boldsymbol{x}\in\mathcal{D}} \mathrm{Sim}(f_\theta(\boldsymbol{x}), g^m_{\boldsymbol{\Theta}}(\boldsymbol{x})) \,, \tag{6}$$

where $\mathrm{Sim}(\cdot,\cdot)$ measures the similarity between the model outputs. While KL-divergence and JS divergence are commonly used for this purpose, they are constrained to comparing models within the same label space. When the PTM and target model possess non-overlapping class sets, alternative techniques that consider statistics of hidden layers [Romero *et al.*, 2015], as well as the relationships between instances [Park *et al.*, 2019], prove to be powerful tools for achieving alignment.

For large models. Fine-tuning, hypothesis transfer, and knowledge distillation have been proven effective in adapting PTMs for small models. Correspondingly, similar ideas are also applied for large models, *e.g.*, distilling large LLMs into smaller scale [Gu *et al.*, 2024], reprogramming the model with extra mapping layers [Chen, 2024], applying lightweight tuning techniques for LLM [Hu *et al.*, 2022]. Additionally, when facing diverse modalities, reusing multiple large models of different modalities for unified recognition becomes popular in the field. For example, LLaVA [Liu *et al.*, 2023] and BLIP [Li *et al.*, 2023] build multi-modal LLMs by learning the projection between frozen visual and textual features.

Pros & Cons. We visualize the steps to reuse PTMs during model training in Figure 4. By leveraging PTMs, the training time is significantly reduced, as the complex foundational learning has already been accomplished. Furthermore, PTMs often lead to improved model performance, especially when fine-tuned for specific tasks, because they have been pre-trained on extensive and diverse datasets. However, potential risks include overfitting the PTM's data characteristics, especially if the target task differs significantly from the tasks the PTM was originally trained on. Finally, the success of the adapted model heavily depends on the quality of the PTM. If the PTM has biases or was trained on non-representative data, these issues can propagate to the adapted model.

Adapt PTMs during model inference

For small models. The advantages of PTMs extend to the model inference stage of the target model. One straightforward approach involves constructing an embedding space using the PTM, resulting in more discriminative features. This enriched space facilitates the creation of linear classifiers and nearest class mean classifier [Mensink *et al.*, 2013], simplifying the process of obtaining the target model. Some recent methods also learn to concatenate middle layers' features [Evci *et al.*, 2022] or learn to weight features from different PTMs [Tu *et al.*, 2023] to enhance the generalization of the features.

Furthermore, model merging methods have been developed to fuse PTMs together. This technique has the potential to flatten the optimization space and increase the likelihood of converging towards an optimal solution [Matena and Raffel, 2022]. Various methods have emerged in this domain, including those emphasizing partial channels during the model merging process [Wortsman *et al.*, 2022]. Some other methods address the model heterogeneity. These approaches involve learning a mapping matrix to rectify the weights of different layers [Singh and Jaggi, 2020], accommodating the diverse characteristics of the models being merged.

For large models. The idea of model merging is also proven effective in LLM inference to enhance its general capability [Yang *et al.*, 2024; Kim *et al.*, 2024; Dang *et al.*, 2025]. Besides, Retrieval-Augmented Generation (RAG) [Lewis *et al.*, 2020] utilizes the external database for retrieval. Facing a new input, it fetches relevant documents from the database and produces responses based on the retrieved information. When an external database is unavailable, methods also explore the LLMs' reasoning ability to self-guide the inference [Brown *et al.*, 2020] by reasoning "step by step".

Pros & Cons. Leveraging PTMs during inference can reduce the need for expensive model retraining, thus saving computational resources and time. However, there are also some drawbacks. While PTMs can be powerful, integrating them effectively with the target model can be challenging, especially when concatenating or combining features from different layers. This may require careful fine-tuning and architecture adjustments. Besides, although PTMs can simplify some aspects of inference, the added complexity of using intermediate layer features or constructing chain of thoughts may introduce additional computation, potentially increasing inference time compared to simpler models.

3.3 Other Topics

Apart from the above topics in model reuse, there are also related fields focusing on other aspects of model reuse. These topics are not fully independent of model selection and adaptation, which partially share the same goal.

Model Assembly. Rather than treating each PTM as a whole, model reuse can operate at a finer granularity by assembling models from components such as network blocks [Yang *et al.*, 2022; Pfeiffer *et al.*, 2024; Guo *et al.*, 2023] or parameter-efficient tuning modules [Hu *et al.*, 2024]. This approach enables fine-grained model selection targets, where reusable parts are identified and recombined to build models tailored to specific tasks. Some methods also modularize networks or extract key functional units to improve interpretability and flexibility [Wang *et al.*, 2022; Shi *et al.*, 2024].

Model Representation Learning [Schürholt *et al.*, 2022; Unterthiner *et al.*, 2020] vectorizes PTMs to encode their characteristics, forming hyper-representations that allow model selection without direct access to the original model parameters. These representations can be learned based on intrinsic permutation-invariant properties of a PTM and can even be decoded back into model weights [Schürholt *et al.*, 2024]. Learned model representations serve multiple purposes: they act as model specifications within a model zoo, assist in efficient retrieval, and serve as auxiliary modalities during the adaptation process [Zhou *et al.*, 2024].

Model Compression. Given the computational demands of large PTMs, model compression aims to reduce model size while preserving functionality. Techniques such as pruning [Cheng *et al.*, 2024] and quantization [Zhou *et al.*, 2018] are used to transform large PTMs into compact versions that maintain performance while reducing inference costs.

Model Repair and Editing. Model repair corrects erroneous behaviors in a PTM without retraining from scratch [Luo *et al.*, 2023], often through targeted weight modifications, counterfactual learning, or fine-grained intervention. Model editing [Mitchell *et al.*, 2022; Fang *et al.*, 2024], on the other hand, involves modifying a model's knowledge base or decision boundaries to reflect new information or correct biases. Unlike standard fine-tuning, model editing aims to localize changes while maintaining overall model consistency.

Managing LLMs. Managing model repositories such as "LLM Market" requires dynamic specification of model traits, rapid adaptation to new data, and efficient fusion and configuration of multiple models [Shen *et al.*, 2024; Prabhu *et al.*, 2024; Zhou *et al.*, 2025; Zhang *et al.*, 2024]. Methods have been developed to learn representations of model characteristics and add semantic specifications to guide model selection. Efficient evaluation frameworks assess language understanding, reasoning, and task proficiency of LLMs, facilitating benchmarking and optimization for real-world deployment [Polo *et al.*, 2024; Zhong *et al.*, 2025; Saranathan *et al.*, 2024].

Model Relationship Discovery. PTM relationship discovery aims to trace the relationship between models, providing helpful information in model zoo management. Neural Lineage [Yu and Wang, 2024] predicts which parent model a child model has been fine-tuned from, and [Horwitz *et al.*, 2025] formulates this task in an unsupervised manner.

4 Future Directions

Coupling of Selection and Adaptation. The generalization ability of an adapted model depends not only on the selected PTM but also on the adaptation strategy employed. Thus, PTM selection and adaptation are inherently coupled [Arango *et al.*, 2024]. While some methods, such as transferability estimation, attempt to improve selection, they often fail to account for the mismatches between fine-tuning and advanced adaptation techniques. Future research should focus on joint optimization of selection and adaptation to enhance end-to-end performance.

Diversity and Strength of the Model Zoo. The success of model reuse depends on the quality, diversity, and generalization of PTMs in the model zoo. A well-curated collection improves the chance of selecting suitable models—even at fine-grained levels such as instances or tokens. Therefore, improving the curation and management of model zoos is essential for maximizing the impact and utility of model reuse [Castaño *et al.*, 2023; Dong *et al.*, 2023].

Universal Adaptation Methods. Adaptation methods must bridge heterogeneity between the target task and the selected PTM while remaining universally applicable across different models and tasks. However, most adaptation techniques are task-specific and require extensive hyper-parameter tuning, making them difficult to generalize. Developing standardized, flexible, and efficient adaptation strategies is essential for improving model reuse in diverse settings.

Model Collaboration. Instead of selecting a single PTM, aggregating multiple PTMs to collaborate in parallel or sequentially can enhance model robustness and accuracy [Feng *et al.*, 2025]. In this way, complex target tasks can often be decomposed into multiple sub-tasks, each handled by a different PTM with distinct expertise. There are several ways to achieve this goal. Firstly, models can collaborate sequentially—small models can tackle easy tasks, while large models can tackle what small models cannot handle. Furthermore, some tasks can be decomposed into sub-tasks, where different PTMs handle each step sequentially. Optimizing the order and dependencies of PTMs is crucial, considering task complexity, computational cost, and adaptation requirements [Chen and Varoquaux, 2024]. Secondly, multiple models can collaborate in a parallel manner, where multiple PTMs contribute jointly to task performance through techniques such as weighted averaging or model merging [Wortsman *et al.*, 2022; Stoica *et al.*, 2023]. However, effectively combining PTMs with varying strengths remains a challenge, requiring proper ensemble strategies [Fu *et al.*, 2025]. Lastly, modules of different PTMs can also collaborate together to enhance prediction robustness [Wang *et al.*, 2024; Li *et al.*, 2025].

Model Market and Dock System Construction. As model reuse continues to gain traction, the seamless deployment of reusable models requires a robust, containerized system. Systems like Learnware [Zhou, 2016; Zhou and Tan, 2024] and Beimingwu [Tan *et al.*, 2024] have paved the way for efficient model reuse, offering quick and easy application. However, beyond containerization, several critical topics in the context of the model market remain underexplored. For instance, managing the model zoo to facilitate smoother model reuse, as well as determining appropriate pricing strategies for models, are areas that warrant further research and development.

Privacy-Preserving and Robust PTM Representations. The effectiveness of PTM selection and adaptation often relies on access to large, labeled datasets, which may not always be feasible due to privacy concerns or data scarcity. Future work should explore privacy-preserving PTM representations and methods that require minimal task-specific data [Lei *et al.*, 2024]. In particular, large PTMs may produce nearly identical outputs on few-shot samples, making it difficult to distinguish their capabilities. Developing task-aware probing mechanisms that can effectively differentiate PTMs under low-data conditions is an important research direction.

Multi-Objective Model Reuse. In the real world, apart from vanilla measuring the performance on target tasks, there are other important targets, *e.g.*, adapting efficiency [Houlsby *et al.*, 2019] that require fast model reuse and carbon emission [Lacoste *et al.*, 2019] that recurring low computation cost. These multiple objects may also evolve in the learning process, posing more challenging targets to model reuse.

5 Conclusions

Model reuse has become a common paradigm in modern machine learning, enabling the efficient utilization of PTMs to enhance performance while reducing computational and data requirements. This survey provides a unifying perspective on model reuse, categorizing methods based on PTM selection and adaptation, and exploring its applications across both small specialized PTMs and large foundation models. By outlining a comprehensive taxonomy, discussing emerging trends, and identifying open research questions, this survey aims to provide valuable insights to drive future research and foster the adoption of model-centric AI development.

Acknowledgments

This work is supported by the National Key R&D Program of China (2024YFE0202800), NSFC (62376118), Fundamental Research Funds for the Central Universities (14380021), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[Achille *et al.*, 2019] A. Achille, M. Lam, R. Tewari, A. Ravichandran, S. Maji, C. C. Fowlkes, S. Soatto, and P. Perona. Task2vec: Task embedding for meta-learning. In *ICCV*, 2019.

[Arango *et al.*, 2024] S. P. Arango, F. Ferreira, A. Kadra, F. Hutter, and J. Grabocka. Quick-tune: Quickly learning which pretrained model to finetune and how. In *ICLR*, 2024.

[Bommasani *et al.*, 2021] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021.

[Brown *et al.*, 2020] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020.

[Castaño *et al.*, 2023] J. Castaño, S. Martínez-Fernández, X. Franch, and J. Bogner. Exploring the carbon footprint of hugging face's ml models: A repository mining study. In *ESEM*, 2023.

[Chen and Varoquaux, 2024] L. Chen and G. Varoquaux. What is the role of small models in the LLM era: A survey. *CoRR*, abs/2409.06857, 2024.

[Chen *et al.*, 2022] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *NeurIPS*, 2022.

[Chen, 2024] P.-Y. Chen. Model reprogramming: Resource-efficient cross-domain machine learning. In *AAAI*, 2024.

[Cheng *et al.*, 2024] H. Cheng, M. Zhang, and J. Q. Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *TPAMI*, 2024.

[Dang *et al.*, 2025] X. Dang, C. Baek, K. Wen, Z. Kolter, and A. Raghunathan. Weight ensembling improves reasoning in language models. *CoRR*, abs/2504.10478, 2025.

[Ding *et al.*, 2022] Y.-X. Ding, X.-Z. Wu, K. Zhou, and Z.-H. Zhou. Pre-trained model reusability evaluation for small-data transfer learning. In *NeurIPS*, 2022.

[Dong *et al.*, 2023] Q. Dong, F. Zhou, N. Kang, C. Xie, S. Zhang, J. Li, H. Peng, and Z. Li. Damix: exploiting deep autoregressive model zoo for improving lossless compression generalization. In *AAAI*, 2023.

[Evci *et al.*, 2022] U. Evci, V. Dumoulin, H. Larochelle, and M. C. Mozer. Head2toe: Utilizing intermediate representations for better transfer learning. In *ICML*, 2022.

[Fang *et al.*, 2024] J. Fang, H. Jiang, K. Wang, Y. Ma, S. Jie, X. Wang, X. He, and T.-S. Chua. Alphaedit: Null-space constrained knowledge editing for language models. *CoRR*, abs/2410.02355, 2024.

[Feng *et al.*, 2025] S. Feng, W. Ding, A. Liu, Z. Wang, W. Shi, Y. Wang, Z. Shen, X. Han, H. Lang, C.-Y. Lee, et al. When one llm drools, multi-llm collaboration rules. *CoRR*, abs/2502.04506, 2025.

[Frick *et al.*, 2025] E. Frick, C. Chen, J. Tennyson, T. Li, W.-L. Chiang, A. N. Angelopoulos, and I. Stoica. Prompt-to-leaderboard. *CoRR*, abs/2502.14855, 2025.

[Fu *et al.*, 2025] J. Fu, Y. Jiang, J. Chen, J. Fan, X. Geng, and X. Yang. Speculative ensemble: Fast large language model ensemble via speculation. *CoRR*, abs/2502.01662, 2025.

[Gu *et al.*, 2024] Y. Gu, L. Dong, F. Wei, and M. Huang. Minillm: Knowledge distillation of large language models. In *ICLR*, 2024.

[Guha *et al.*, 2024] N. Guha, M. Chen, T. Chow, I. Khare, and C. Re. Smoothie: Label free language model routing. *NeurIPS*, 2024.

[Guo and Gelfand, 1992] H. Guo and S. B. Gelfand. Classification trees with neural network feature extraction. In *CVPR*, 1992.

[Guo *et al.*, 2023] L.-Z. Guo, Z. Zhou, Y.-F. Li, and Z.-H. Zhou. Identifying useful learnwares for heterogeneous label spaces. In *ICML*, 2023.

[Gupta and Kembhavi, 2023] T. Gupta and A. Kembhavi. Visual programming: Compositional visual reasoning without training. In *CVPR*, 2023.

[He *et al.*, 2016] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Hinton *et al.*, 2015] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[Horwitz *et al.*, 2025] E. Horwitz, A. Shul, and Y. Hoshen. Unsupervised model tree heritage recovery. In *ICLR*, 2025.

[Houlsby *et al.*, 2019] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for NLP. In *ICML*, 2019.

[Hu *et al.*, 2022] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[Hu *et al.*, 2024] J. Hu, J. Gao, J. Ye, Y. Gao, X. Wang, Z. Feng, and M. Song. Model lego: Creating models like disassembling and assembling building blocks. In *NeurIPS*, 2024.

[Jacobs *et al.*, 1991] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 1991.

[Jitkrittum *et al.*, 2025] W. Jitkrittum, H. Narasimhan, A. S. Rawat, J. Juneja, Z. Wang, C.-Y. Lee, P. Shenoy, R. Panigrahy, A. K. Menon, and S. Kumar. Universal llm routing with correctness-based representation. In *ICLR Workshop*, 2025.

[Kim *et al.*, 2024] S. Kim, D. Kim, C. Park, W. Lee, W. Song, Y. Kim, H. Kim, Y. Kim, H. Lee, J. Kim, et al. Solar 10.7 b: Scaling large language models with simple yet effective depth up-scaling. In *NAACL*, 2024.

[Kundu *et al.*, 2020] J. N. Kundu, N. Venkat, R. V. Babu, et al. Universal source-free domain adaptation. In *CVPR*, 2020.

[Kuzborskij and Orabona, 2017] I. Kuzborskij and F. Orabona. Fast rates by transferring from auxiliary hypotheses. *MLJ*, 2017.

[Lacoste *et al.*, 2019] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres. Quantifying the carbon emissions of machine learning. *CoRR*, abs/1910.09700, 2019.

[Lei *et al.*, 2024] H.-Y. Lei, Z.-H. Tan, and Z.-H. Zhou. On the ability of developers' training data preservation of learnware. In *NeurIPS*, 2024.

[Lewis *et al.*, 2020] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*, 2020.

[Li *et al.*, 2018] X. Li, Y. Grandvalet, and F. Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *ICML*, 2018.

[Li *et al.*, 2023] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

[Li *et al.*, 2025] W. Li, Y. Lin, M. Xia, and C. Jin. Rethinking mixture-of-agents: Is mixing different large language models beneficial? *CoRR*, abs/2502.00674, 2025.

[Liang *et al.*, 2022] J. Liang, D. Hu, J. Feng, and R. He. Dine: Domain adaptation from single and multiple black-box predictors. In *CVPR*, 2022.

[Liu *et al.*, 2023] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *NeurIPS*, 2023.

[Lu *et al.*, 2024] K. Lu, H. Yuan, R. Lin, J. Lin, Z. Yuan, C. Zhou, and J. Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. In *NAACL*, 2024.

[Luo *et al.*, 2023] J. Luo, Z. Wang, C. H. Wu, D. Huang, and F. De la Torre. Zero-shot model diagnosis. In *CVPR*, 2023.

[Luo *et al.*, 2024] X. Luo, Z. Deng, B. Yang, and M. Y. Luo. Pretrained language models in medicine: A survey. *Artificial Intelligence in Medicine*, 2024.

[Matena and Raffel, 2022] M. Matena and C. Raffel. Merging models with fisher-weighted averaging. In *NeurIPS*, 2022.

[Menon and Vondrick, 2023] S. Menon and C. Vondrick. Visual classification via description from large language models. In *ICLR*, 2023.

[Mensink *et al.*, 2013] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *TPAMI*, 2013.

[Mitchell *et al.*, 2022] E. Mitchell, C. Lin, A. Bosselut, C. D. Manning, and C. Finn. Memory-based model editing at scale. In *ICML*, 2022.

[Mitchell, 1997] T. Mitchell. *Machine learning*, volume 1. 1997.

[Miyato *et al.*, 2018] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[Mohammadshahi *et al.*, 2024] A. Mohammadshahi, A. Shaikh, and M. Yazdani. Leeroo orchestrator: Elevating llms performance through model integration. *CoRR*, abs/2401.13979, 2024.

[Muqeeth *et al.*, 2024] M. Muqeeth, H. Liu, Y. Liu, and C. Raffel. Learning to route among specialized experts for zero-shot generalization. In *ICML*, 2024.

[Nguyen *et al.*, 2020] C. V. Nguyen, T. Hassner, C. Archambeau, and M. Seeger. Leep: A new measure to evaluate transferability of learned representations. In *ICML*, 2020.

[Ong *et al.*, 2025] I. Ong, A. Almahairi, V. Wu, W.-L. Chiang, T. Wu, J. E. Gonzalez, M. W. Kadous, and I. Stoica. Routellm: Learning to route llms from preference data. In *ICLR*, 2025.

[Pándy *et al.*, 2022] M. Pándy, A. Agostinelli, J. R. R. Uijlings, V. Ferrari, and T. Mensink. Transferability estimation using bhattacharyya class separability. In *CVPR*, 2022.

[Park *et al.*, 2019] W. Park, D. Kim, Y. Lu, and M. Cho. Relational knowledge distillation. In *CVPR*, 2019.

[Pfeiffer *et al.*, 2020] J. Pfeiffer, A. Rücklé, C. Poth, A. Kamath, I. Vulić, S. Ruder, K. Cho, and I. Gurevych. Adapterhub: A framework for adapting transformers. In *EMNLP (Demos)*, 2020.

[Pfeiffer *et al.*, 2024] J. Pfeiffer, S. Ruder, I. Vulić, and E. Ponti. Modular deep learning. *TMLR*, 2024.

[Polo *et al.*, 2024] F. M. Polo, L. Weber, L. Choshen, Y. Sun, G. Xu, and M. Yurochkin. tinybenchmarks: evaluating llms with fewer examples. In *ICML*, 2024.

[Prabhu *et al.*, 2024] A. Prabhu, V. Udandarao, P. Torr, M. Bethge, A. Bibi, and S. Albanie. Efficient lifelong model evaluation in an era of rapid progress. In *NeurIPS*, 2024.

[Pratt *et al.*, 2023] S. Pratt, I. Covert, R. Liu, and A. Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *ICCV*, 2023.

[Romero *et al.*, 2015] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[Saranathan *et al.*, 2024] G. Saranathan, M. P. Alam, J. Lim, S. Bhattacharya, S. Y. Wong, M. Foltin, and C. Xu. Dele: Data efficient llm evaluation. In *ICLR Workshop*, 2024.

[Schürholt *et al.*, 2022] K. Schürholt, D. Taskiran, B. Knyazev, X. Giró-i Nieto, and D. Borth. Model zoos: A dataset of diverse populations of neural network models. *NeurIPS*, 2022.

[Schürholt *et al.*, 2024] K. Schürholt, M. W. Mahoney, and D. Borth. Towards scalable and versatile weight space learning. In *ICML*, 2024.

[Shen *et al.*, 2024] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. In *NeurIPS*, 2024.

[Shi *et al.*, 2024] B. Shi, S. Xia, X. Yang, H. Chen, Z. Kou, and X. Geng. Building variable-sized models via learngene pool. In *AAAI*, 2024.

[Singh and Jaggi, 2020] S. P. Singh and M. Jaggi. Model fusion via optimal transport. In *NeurIPS*, 2020.

[Smith *et al.*, 2023] J. S. Smith, Y.-C. Hsu, L. Zhang, T. Hua, Z. Kira, Y. Shen, and H. Jin. Continual diffusion: Continual customization of text-to-image diffusion with c-lora. *CoRR*, abs/2304.06027, 2023.

[Song *et al.*, 2019] J. Song, Y. Chen, X. Wang, C. Shen, and M. Song. Deep model transferability from attribution maps. In *NeurIPS*, 2019.

[Stoica *et al.*, 2023] G. Stoica, D. Bolya, J. Bjorner, T. Hearn, and J. Hoffman. Zipit! merging models from different tasks without training. In *CVPR*, 2023.

[Tan *et al.*, 2024] Z.-H. Tan, J.-D. Liu, X.-D. Bi, P. Tan, Q.-C. Zheng, H.-T. Liu, Y. Xie, X.-C. Zou, Y. Yu, and Z.-H. Zhou. Beimingwu: A learnware dock system. In *KDD*, 2024.

[Tan *et al.*, 2025] Z.-H. Tan, Z.-C. Zhao, H.-Y. Shi, X.-Y. Zhang, P. Tan, Y. Yu, and Z.-H. Zhou. Learnware of language models: Specialized small language models can do big. *CoRR*, abs/2505.13425, 2025.

[Tran *et al.*, 2019] A. T. Tran, C. V. Nguyen, and T. Hassner. Transferability and hardness of supervised classification tasks. In *ICCV*, 2019.

[Tu *et al.*, 2023] C.-H. Tu, Z. Mai, and W.-L. Chao. Visual query tuning: Towards effective usage of intermediate representations for parameter and memory efficient transfer learning. In *CVPR*, 2023.

[Unterthiner *et al.*, 2020] T. Unterthiner, D. Keysers, S. Gelly, O. Bousquet, and I. Tolstikhin. Predicting neural network accuracy from weights. *CoRR*, abs/2002.11448, 2020.

[Wang *et al.*, 2022] Q.-F. Wang, X. Geng, S.-X. Lin, S.-Y. Xia, L. Qi, and N. Xu. Learngene: From open-world to your learning task. In *AAAI*, 2022.

[Wang *et al.*, 2024] J. Wang, J. Wang, B. Athiwaratkun, C. Zhang, and J. Zou. Mixture-of-agents enhances large language model capabilities. *CoRR*, abs/2406.04692, 2024.

[Wolf, 2019] T. Wolf. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

[Wortsman *et al.*, 2022] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. G. Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, and L. Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*, 2022.

[Wu *et al.*, 2019] X.-Z. Wu, S. Liu, and Z.-H. Zhou. Heterogeneous model reuse via optimizing multiparty multiclass margin. In *ICML*, 2019.

[Wu *et al.*, 2024] X. Wu, S. Huang, and F. Wei. Mixture of lora experts. In *ICLR*, 2024.

[Xue *et al.*, 2024] Y. Xue, R. Yang, X. Chen, W. Liu, Z. Wang, and X. Liu. A review on transferability estimation in deep transfer learning. *TAI*, 2024.

[Yang *et al.*, 2022] X. Yang, D. Zhou, S. Liu, J. Ye, and X. Wang. Deep model reassembly. In *NeurIPS*, 2022.

[Yang *et al.*, 2024] E. Yang, L. Shen, G. Guo, X. Wang, X. Cao, J. Zhang, and D. Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *CoRR*, abs/2408.07666, 2024.

[Ye *et al.*, 2021] H.-J. Ye, D.-C. Zhan, Y. Jiang, and Z.-H. Zhou. Heterogeneous few-shot model rectification with semantic mapping. *TPAMI*, 2021.

[Yi *et al.*, 2024] C. Yi, D.-C. Zhan, and H.-J. Ye. Bridge the modality and capacity gaps in vision-language model selection. *NeurIPS*, 2024.

[Yosinski *et al.*, 2014] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014.

[Yu and Wang, 2024] R. Yu and X. Wang. Neural lineage. In *CVPR*, 2024.

[Zamir *et al.*, 2018] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018.

[Zhang *et al.*, 2023] Y.-K. Zhang, T.-J. Huang, Y.-X. Ding, D.-C. Zhan, and H.-J. Ye. Model spider: Learning to rank pre-trained models efficiently. In *NeurIPS*, 2023.

[Zhang *et al.*, 2024] Q. Zhang, F. Lyu, X. Liu, and C. Ma. Collaborative performance prediction for large language models. In *EMNLP*, 2024.

[Zhang *et al.*, 2025] Y.-K. Zhang, D.-C. Zhan, and H.-J. Ye. Capability instruction tuning. In *AAAI*, 2025.

[Zhao *et al.*, 2023] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. *CoRR*, abs/2303.18223, 2023.

[Zheng *et al.*, 2023] H. Zheng, L. Shen, A. Tang, Y. Luo, H. Hu, B. Du, and D. Tao. Learn from model beyond fine-tuning: A survey. *CoRR*, abs/2310.08184, 2023.

[Zhong *et al.*, 2025] X.-X. Zhong, C. Yi, and H.-J. Ye. Efficient evaluation of large language models via collaborative filtering. *CoRR*, abs/2504.08781, 2025.

[Zhou and Jiang, 2004] Z.-H. Zhou and Y. Jiang. Nec4. 5: Neural ensemble based c4. 5. *TKDE*, 2004.

[Zhou and Tan, 2024] Z.-H. Zhou and Z.-H. Tan. Learnware: small models do big. *Sci. China Inf. Sci.*, 2024.

[Zhou *et al.*, 2018] Y. Zhou, S.-M. Moosavi-Dezfooli, N.-M. Cheung, and P. Frossard. Adaptive quantization for deep neural network. In *AAAI*, 2018.

[Zhou *et al.*, 2024] A. Zhou, C. Finn, and J. Harrison. Universal neural functionals. *CoRR*, abs/2402.05232, 2024.

[Zhou *et al.*, 2025] J. P. Zhou, C. K. Belardi, R. Wu, T. Zhang, C. P. Gomes, W. Sun, and K. Q. Weinberger. On speeding up language model evaluation. In *ICLR*, 2025.

[Zhou, 2016] Z.-H. Zhou. Learnware: on the future of machine learning. *Frontiers Comput. Sci.*, 2016.

[Zhuang *et al.*, 2020] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *Proc. IEEE*, 2020.

[Zohar *et al.*, 2023] O. Zohar, S.-C. Huang, K.-C. Wang, and S. Yeung. Lovm: Language-only vision model selection. *NeurIPS*, 2023.