LEVI: Generalizable Fine-tuning via Layer-wise Ensemble of Different Views

Yuji Roh [1] **Qingyun Liu** [2] **Huan Gui** [2] **Zhe Yuan** [1] **Yujin Tang** [2]
Steven E. Whang [3] **Liang Liu** [1] **Shuchao Bi** [1] **Lichan Hong** [2] **Ed H. Chi** [2] **Zhe Zhao** [2]

Abstract

Fine-tuning is becoming widely used for leveraging the power of pre-trained foundation models in new downstream tasks. While there are many successes of fine-tuning on various tasks, recent studies have observed challenges in the generalization of fine-tuned models to unseen distributions (i.e., out-of-distribution; OOD). To improve OOD generalization, some previous studies identify the limitations of fine-tuning data and regulate fine-tuning to preserve the general representation learned from pre-training data. However, potential limitations in the pre-training data and models are often ignored. In this paper, we contend that overly relying on the pre-trained representation may hinder fine-tuning from learning essential representations for downstream tasks and thus hurt its OOD generalization. It can be especially catastrophic when new tasks are from different (sub)domains compared to pre-training data. To address the issues in both pre-training and fine-tuning data, we propose a novel generalizable fine-tuning method LEVI (**L**ayer-wise **E**nsemble of different **VI**ews), where the pre-trained model is adaptively ensembled layer-wise with a small task-specific model, while preserving its efficiencies. By combining two complementing models, LEVI effectively suppresses problematic features in both the fine-tuning data and pre-trained model and preserves useful features for new tasks. Broad experiments with large language and vision models show that LEVI greatly improves fine-tuning generalization via emphasizing different views from fine-tuning data and pre-trained features.

[1]Google Inc, Mountain View, CA, USA [2]Google DeepMind, Mountain View, CA, USA [3]School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. Correspondence to: Yuji Roh <yujiroh@google.com>.

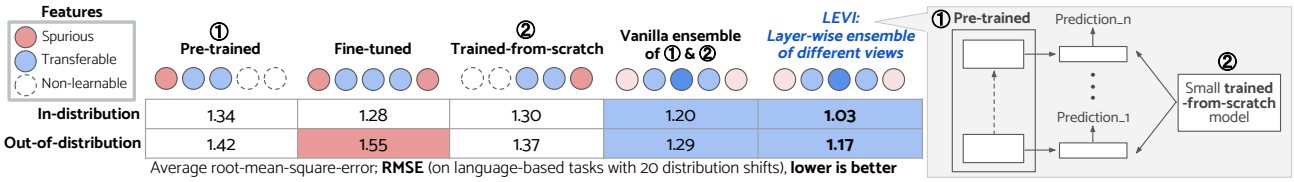
Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. Introduction

Recent breakthroughs in foundation models make various high-quality pre-trained models available to the public (Devlin et al., 2019; Radford et al., 2021), and the *fine-tuning* paradigm has become a prevalent approach for leveraging pre-trained models' power in new downstream tasks. By tailoring the pre-trained features to align with the characteristics of the new tasks, fine-tuning shows promising performances in various scenarios, including natural language and computer vision (Kornblith et al., 2019; Guu et al., 2020).

Despite many success stories of fine-tuning, recent studies have observed that fine-tuned models often fail to ensure consistent generalization across new distributions (i.e., *out-of-distribution*; OOD) at deployment time (Bommasani et al., 2021; Kumar et al., 2022b), where OOD samples come from a different distribution than the data the model was fine-tuned on. Unlike traditional OOD generalization studies, fine-tuning pre-trained models faces unique challenges in improving OOD generalization, including computational costs of handling large models and the lack of access to pre-training data, which may already have inherent issues.

Although several algorithms have been recently proposed to enhance fine-tuning generalization (Kumar et al., 2022a;b; Wortsman et al., 2022b; Tian et al., 2023), most of them do not consider the inherent limitations in the pre-trained models. Specifically, many previous works implicitly assume that 1) the pre-trained models already have good enough features for the new tasks and that 2) any problems with generalization stem from the downstream (fine-tuning) data. As a result, these algorithms mainly focus on preserving the original pre-trained features and avoiding overfitting to the fine-tuning data's problems, such as spurious features – informative during training, but not useful (transferable) in general. However, such assumptions in prior works may not be true in practice, and pre-trained features can also have inherent issues. For example, even if there is no spurious feature in the fine-tuning data, some of the features in the pre-trained model can be improperly used in the downstream tasks (Xue et al., 2023), e.g., pre-trained demographic features in language models can be wrongly used in new ranking systems. Moreover, pre-trained features may not have all the important representations for the new tasks (Kang



	① Pre-trained	Fine-tuned	② Trained-from-scratch	Vanilla ensemble of ① & ②	LEVI: Layer-wise ensemble of different views
In-distribution	1.34	1.28	1.30	1.20	**1.03**
Out-of-distribution	1.42	1.55	1.37	1.29	**1.17**

Average root-mean-square-error; **RMSE** (on language-based tasks with 20 distribution shifts), **lower is better**

Figure 1. When both pre-trained features and fine-tuning data have inherent problems like spurious features, they can jointly affect the OOD generalization ability of a resulting fine-tuned model. Indeed, we observe that the OOD performance of the fine-tuned model is worse (red color in the table) than both the pre-trained and trained-from-scratch (i.e., randomly initialized then trained on fine-tuning data) models, where we 1) fine-tune a pre-trained language model (T5x) on various downstream tasks (movie and product recommendations) and 2) test on 20 distribution shifts (e.g., subpopulation and time shifts). To address this issue, our key idea is to *separately leverage different views* from a pre-trained model and a trained-from-scratch model via *layer-wise ensemble* to reduce the impact of problematic features while preserving necessary ones. Compared to the vanilla ensemble of such two complementing models (fourth column), LEVI further improves both ID and OOD performances while preserving training and inference efficiencies – see framework details in Sec. 4.

et al., 2023; Bommasani et al., 2021); thus, simply preserving and relying on pre-trained representations may not be enough to learn essential ones for the new tasks.

Therefore, we aim to *mitigate the limitations from both pre-trained features and fine-tuning data while maintaining necessary features* to improve fine-tuning generalization. To systematically understand the problem, we use the spurious feature problem as a key example, as it is one of the most important factors causing OOD generalization failure. Notably, Fig. 1 shows that when both the pre-trained model and fine-tuning data contain spurious features, the generalization ability of the fine-tuned model can be even worse than the pre-trained and trained-from-scratch (i.e., randomly initialized then trained on fine-tuning data) models, as the fine-tuned model can be affected by more spurious features than the other two. We provide other real-world cases in Sec. 5. Here, the main challenge in the fine-tuning scenario is the lack of information about spurious features, especially from pre-training data that is unavailable during fine-tuning.

Our key insight to tackle this challenge is using multiple views from both pre-training and fine-tuning data to compensate each other's weaknesses and keep useful features. We thus propose LEVI, a **L**ayer-wise **E**nsemble for generalizable fine-tuning, which adaptively combines different **VI**ews from a large pre-trained model with those from a small, trained-from-scratch model to implicitly mitigate the impacts of spurious features and preserve necessary features, as described in the rightmost part of Fig. 1. The fundamental insight of LEVI is to harness the different views offered by such two complementary models. Compared to traditional fine-tuning that incrementally updates a pre-trained model on the fine-tuning data, we separate and jointly emphasize the complementary information from both sides. Also, as in Fig. 1, LEVI further improves performance and efficiency beyond simple prediction ensembles by tightly integrating the intermediate layers of the pre-trained model, which also offer different views (e.g., early layers are more general, and the later layers are more specific) – see details in Sec. 4.

We note that there are several recent ensemble-based approaches for fine-tuning generalization (Kumar et al., 2022a; Wortsman et al., 2022b) that only utilize the pre-trained model variants, but we are the first to leverage the information from the trained-from-scratch models to mitigate the inherent problems from both pre-trained model and fine-tuning data and to generate new representations – see detailed comparisons in Secs. 2 & 5. Especially for training from scratch, LEVI uses a relatively small yet task-specialized model, which allows to decrease the computational costs of ensembling while giving additional benefits to effectively learn task-specialized features for new tasks.

Extensive experiments on language-based recommendation and computer vision tasks show that LEVI greatly improves the generalization ability of fine-tuning. We observe the state-of-the-art results in various OOD scenarios, including subpopulation, time, and domain shifts. We also show that our approach is more efficient than most existing ensemble-based generalization approaches. In addition, various ablation studies and hyperparameter analyses help to understand LEVI's behaviors. Finally, LEVI can be gracefully merged with efficient fine-tuning methods such as LoRA (Hu et al., 2022), resulting in further improved training efficiency.

Summary of Contributions: (1) We reveal the importance of addressing inherent problems of not only fine-tuning data, but also pre-trained models via theoretical and empirical insights. (2) Based on such insights, we propose a novel layer-wise ensemble LEVI for fine-tuning OOD generalization, which synergistically combines different views from two complementing models. (3) We show the value of leveraging trained-from-scratch models in mitigating the limitations of pre-trained models. (4) LEVI largely improves OOD generalization in both language and vision models, while preserving training and inference efficiencies.

2. Related Work

OOD Generalization in the Traditional Literature

Among the broad model generalization issues, making the

model more robust to various OOD scenarios becomes indispensable for AI deployment (Shen et al., 2021a). Although there are many promising directions (Finn et al., 2017; Li et al., 2018; D'Innocente & Caputo, 2019; Carlucci et al., 2019; Raghunathan et al., 2020; Roh et al., 2023), many traditional studies do not consider the challenges in large models (e.g., inherent problems in the pre-trained features and scalability) or assume to access pre-training or unlabeled test data. In comparison, our work improves generalization in the fine-tuning pre-trained model paradigm, where 1) the given models may already have inherent issues, 2) the training and inference efficiency becomes more important, and 3) additional information on the pre-training or deployment (OOD) data is unavailable. We leave a more detailed discussion on the traditional OOD literature in Sec. A.

OOD Generalization in the Fine-tuning Paradigm
With the rapid growth of large pre-trained models, there is an emerging focus on OOD generalization in fine-tuning.

As the pre-training data is known to cover large and diverse distributions, many recent approaches aim to preserve the pre-trained features for improving the generalization (Kumar et al., 2022b; Tian et al., 2023). For example, Kumar et al. (2022b) propose a two-step approach that first linear probes the last layer and then fine-tunes all the parameters so as to mitigate the feature distortion during fine-tuning. Another work (Tian et al., 2023) designs a constrained bi-level optimization to minimally change the pre-trained features. In comparison, we consider the inherent problems in the pre-trained model itself, including the limited features for supporting downstream tasks and spurious correlations between pre-trained features and the target labels. Furthermore, to our understanding, we are the first to reveal the value of using trained-from-scratch models in mitigating inherent issues of pre-trained models.

Among the fine-tuning generalization studies, recent ensemble approaches are the most relevant to our work as they utilize the information from multiple models for generalization and show promising improvements in computer vision tasks with various OOD scenarios (Kumar et al., 2022a; Wortsman et al., 2022b; Pagliardini et al., 2023). For example, Kumar et al. (2022a) average the outputs of the standard fine-tuned and robust-aware fine-tuned models. Another line of studies (Wortsman et al., 2022b; Rame et al., 2023; Wortsman et al., 2022a) ensemble the model weights of the pre-trained model variants to gather diverse information – e.g., ensembling fine-tuned and zero-shot models or averaging multiple fine-tuned weights. Compared to these studies, LEVI is the first work to emphasize the importance of separately treating the complementary views in pre-trained features and fine-tuning data, so as to also mitigate the inherent problems in the pre-trained features. Another ensemble-based approach (Pagliardini et al., 2023) trains multiple models to have disagreed predictions on the OOD distributions by accessing the unlabeled test data. In contrast, our work does not assume any information on the test OOD distribution. Moreover, LEVI does not require more than one large model, preserving both training and inference efficiencies.

In addition, a recent work (Xue et al., 2023) considers the spurious correlations in pre-trained models that can harm the group robustness and mitigates their impacts by utilizing a group-balanced dataset. However, this work assumes to have the information of spurious correlation in advance, which may not be available in real-world applications, especially when the pre-trained features and the target tasks become complex. In contrast, we do not use any prior knowledge of spurious correlations in both the model and data.

3. When Fine-Tuning Fails to Generalize

To improve fine-tuning generalization, we first explain how the inherent problems in the pre-trained model and fine-tuning data can jointly harm model generalization (Sec. 3.1) and discuss the limitations of previous approaches (Sec. 3.2). In this paper, we use the following settings and notations.

Settings. We consider three data distributions each for pre-training, fine-tuning, and testing. Pre-training data and test data are not available, and we only have fine-tuning data and pre-trained model features. When we refer to training data, we mean the fine-tuning data. We consider the fine-tuning distribution as *in-distribution (ID)*, and the test distribution as *out-of-distribution (OOD)*. Thus, ID data represents the samples that the model has been trained, and OOD data represents unfamiliar samples not seen during training.

Notations. Let $\mathrm{x} \in \mathbb{X}$, $\mathrm{y} \in \mathbb{Y}$, and $\hat{\mathrm{y}} \in \mathbb{Y}$ be the input feature, true label, and predicted label, respectively. Let D be the data distribution. Let $\boldsymbol{w}$ be the model weights. The empirical loss is given by $L(\boldsymbol{w})=\frac{1}{m}\sum_i \ell(\mathrm{y}_i, \hat{\mathrm{y}}_i)$, where m is the number of data samples, and $\ell(\cdot)$ is the loss function.

3.1. Theoretical Backgrounds

We first provide theoretical backgrounds that show the fine-tuned model can suffer from the spurious features from both pre-trained models and fine-tuning data. Here, spurious features are defined as features that are useful to increase training (in-distribution) accuracy, but not transferable at deployment with new distributions. These features are known to hurt model generalization (Sagawa et al., 2020; McCoy et al., 2019). We note that the pre-trained models can have other inherent problems like demographic bias, but our work focuses on the issue directly affecting model generalization.

Traditionally, the influence of spurious features *in training data* on model performance has received much attention. For example, Nagarajan et al. (2021) show that when there

are statistical or geometric relationships between spurious features and labels, the empirical risk minimization (ERM)-based model can rely on such spurious correlations in prediction, resulting in a worse OOD generalization. Lemma 1 summarizes the simple yet critical theoretical insights of previous work (Nagarajan et al., 2021; Khani & Liang, 2021).

Lemma 1. *ERM-based model training can be affected by spurious features in the training data. Let the training data D has the input features $\mathrm{x} = [x_1, x_2]$, where x_1 is a spurious feature and x_2 is a transferable feature. When we train a model with randomly initialized weights $\boldsymbol{w} := [w_1, w_2]$ on D, the ERM-based trained model will have a non-zero w_1, indicating a spurious correlation was used to predict labels.*

More recently, there is a new focus on investigating the potential harms from spurious correlations embedded *within pre-trained features*. Specifically, Xue et al. (2023) show that even if there is no spurious correlation in the fine-tuning data, the features in the pre-trained model can be spuriously used in the downstream task.

Lemma 2. *(From (Xue et al., 2023)) If a pre-trained model has spurious features for the downstream task, even the fine-tuning data that does not have spurious features may not eliminate the impacts of spurious correlations already embedded in the pre-trained model. Let the training data D have the input features $\mathrm{x}=[0, x_2]$, where x_2 is a transferable feature (i.e., no spurious feature). When we fine-tune a pre-trained model with non-zero weights $\boldsymbol{w} := [w_1, w_2]$ on D, the resulting fine-tuned model will still have non-zero w_1.*

Given the above lemmas, we can consider a case where a pre-trained model and fine-tuning data have different spurious features. Let the model weight $\boldsymbol{w}$ be $[w_1, w_2, w_3]$ and the data feature x be $[x_1, x_2, x_3]$, where x_1 and x_3 are spurious features for the downstream task. Here, a pre-trained model has an inherent spurious correlation with x_1 (i.e., nonzero w_1), and fine-tuning data contains only the spurious feature x_3, but not x_1 (i.e., $[0, x_2, x_3]$). When we train the given pre-trained model on the fine-tuning data, the model training will learn the spurious correlation from x_3 (Lemma 1), while not eliminating the pre-trained spurious correlation from x_1 (i.e., nonzero w_1), as the fine-tuning data is orthogonal to x_1 and cannot affect the model weight w_1 during training (Lemma 2). As a result, we get the next corollary.

Corollary 3. *If both the pre-trained model and fine-tuning data have spurious features to the downstream labels, both spurious features will jointly affect the fine-tuned model.*

Remark 4. *Our work focuses on when spurious features in pre-training and fine-tuning data do not overlap. We note that when the two data share the same spurious features, the problem turns into a traditional spurious correlation study.*

We provide a toy example in Fig. 2 to give more intuition on how both spurious features can jointly affect the fine-tuned model (Corollary 3), using a duck image classification scenario, where 1) the majority of duck images in fine-tuning data contain the white color of ducks, and 2) a given model is pre-trained to learn and focus on general vision features like backgrounds of the image. Here, on the one hand, fine-tuning data may provide a white color bias during training, which can be used as spurious features derived from the data. As we discussed, many previous studies on fine-tuning generalization focus on such issues from fine-tuning data. On the other hand, a pre-trained model itself may try to classify ducks using the background information based on its prior knowledge from pre-training data. Although the background information may be a good feature in general, it can become a spurious feature for this task, as ducks can be in various different places like ponds and grass, which are also accessible to other animals. Hence, the feature may not generalize well to test data containing such OODs.

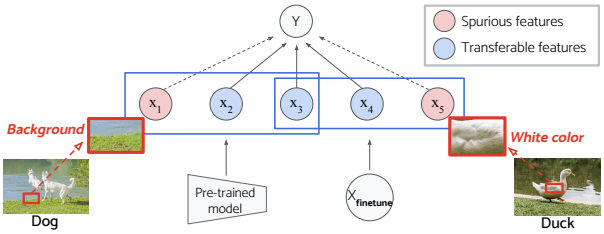


Figure 2. Toy example of a duck classification scenario.

Thus, as the fine-tuned model can be affected by more spurious correlations (e.g., both x_1 and x_5 in Fig. 2) than the pre-trained model and the trained-from-scratch model (i.e., randomly initialized then trained on fine-tuning data), the generalization ability of the fine-tuned model can be even worse than the other two. This insight also aligns well with Fig. 1 – see a more concrete synthetic example in Sec. B and empirical results on real-world benchmarks in Sec. 5.

3.2. Limitation of Previous Approaches

Despite the importance of mitigating the influence of inherent issues on both fronts for enhancing fine-tuning generalization, it has not been actively considered in previous studies. Most existing studies for fine-tuning generalization challenge the long-held assumptions that pre-trained model features are both 1) free from inherent flaws and 2) good enough to support new tasks. Their strategies thus focus on preserving the original pre-trained features during fine-tuning and avoiding overfitting to the fine-tuning data.

However, these assumptions may not hold in reality. First, pre-trained features may contain inherent problems (Bommasani et al., 2021; Xue et al., 2023), which can also limit OOD generalization, as observed above. Also, pre-trained features may not contain all necessary information for new tasks, thus simply preserving the pre-trained features cannot be the best solution to adequately support such new tasks.

4. Framework

We now design a new fine-tuning framework for improving generalization in OODs. We follow the previous theoretical insights, which show the importance of reducing the impact of spurious features from both pre-trained features and fine-tuning data while maintaining necessary features.

Key Intuition. Our main idea is to leverage *different views from different models* to mitigate problems in both pre-trained features and fine-tuning data, while learning essential representations. Despite the importance of addressing spurious correlations during fine-tuning, we usually do not have any information about them, especially those from the pre-training distribution. It is thus hard to explicitly prevent utilizing spurious correlations during fine-tuning. Instead, we implicitly mitigate such spurious correlations by merging complementary information from two very different models: a pre-trained model and a trained-from-scratch model. These models will not be affected by the spurious features from each other, thus we expect to reduce the impact of such features by comparing the complementing information between them. Here, a key difference from conventional fine-tuning is that we separate and jointly emphasize the signals from pre-training and fine-tuning (downstream) data.

Using Ensemble. A key question is how we can effectively combine the distinct information from different models. Our starting point is the idea of model ensemble, which is known to be very effective in combining information from multiple models (Zhang & Ma, 2012). Especially, when the models are diverse and independent, ensembling is a good choice to reduce the model variance, which can decrease the generalization error (Kotu & Deshpande, 2014).

Methodology. Our design focuses on two key objectives: 1) harnessing a multitude of diverse views to maximize the benefits of ensembling beyond simply averaging final model predictions, and 2) maintaining efficiencies in both training and inference phases, which is often a critical issue in ensemble methods. We achieve these objectives by leveraging different views from the models in two directions: *among* the models and *within* the model. First, by ensembling pre-trained and trained-from-scratch models, we can obtain a general view from pre-training data and a task-specialized view from fine-tuning data. Also, by ensembling within the model, we can utilize different information from each intermediate layer (e.g., early layers are more general, and the last layers are more specific) (Yosinski et al., 2014). Note that using multiple intermediate layers is also known to be beneficial in fine-tuning generalization (Evci et al., 2022).

By this intuition, we propose LEVI, a layer-wise ensemble framework for fine-tuning generalization. As described in Fig. 3, LEVI tightly integrates 1) a pre-trained model, which does not need to be updated during fine-tuning, together with

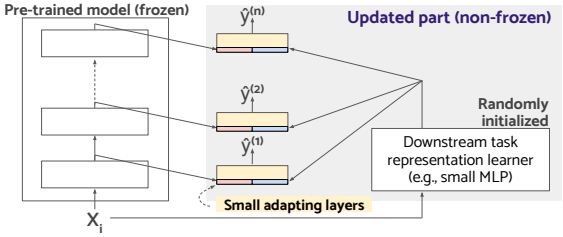


Figure 3. LEVI architecture of using a layer-wise ensemble.

2) a small randomly-initialized model by utilizing adapting layers that concatenate two models. Here, the role of the randomly initialized model is to learn necessary and specialized representations from the fine-tuning data. Also, the model size can be much smaller than the pre-trained model. We then connect the two models with small adapting layers, where their inputs concatenate 1) the pre-trained model's intermediate outputs and 2) the randomly-initialized model final output for each input data. As a result, we can tightly ensemble the information from both pre-trained and trained-from-scratch (i.e., randomly initialized then trained) models. Finally, we set the adapting layer outputs to be the predicted labels and update the model via the following loss function:

$$\min_{\boldsymbol{w}} \frac{1}{m} \cdot \frac{1}{n} \sum_{i}^{m} \sum_{j}^{n} \ell(\mathrm{y}_i, \hat{\mathrm{y}}_i^{(j)}),$$

where m and n are the numbers of training samples and adapting layers, respectively, and $\hat{\mathrm{y}}_i^{(j)}$ is the predicted label from the adapting layer j for the input data i. Note that we consider the equal weights on all $\hat{\mathrm{y}}^{(j)}$, and one can change it as a weighted sum of $\hat{\mathrm{y}}^{(j)}$ – see more discussions in Sec. C.

Benefits. Finally, we highlight two key benefits of LEVI.

Adding a Task-specialized Model: LEVI has the flexibility to use any model architecture as the trained-from-scratch model, which allows for the selection of task-specialized architectures. This flexibility is very beneficial to learn critical features for new tasks effectively, which is particularly notable when the pre-trained features are not enough to support the new tasks. We provide *a concrete example in movie recommendation*, aiming to predict user ratings for movies using a pre-trained language model. LEVI can utilize a multi-layer perceptron (MLP) model with an embedding block as its trained-from-scratch model. Here, such an MLP model is known to be very suitable for handling sparse user and movie information, offering a distinct advantage over conventional language models, thereby enhancing the overall recommendation performances.

Using Multiple Intermediate Layers: Another advantage of LEVI comes from leveraging both the later and early layers of a large model, which contributes to enhancing the trade-off between in-distribution (ID) and out-of-distribution (OOD) performances. In our experiments, we observe notable results indicating that the later layers of the large model

enhance ID performance, while the early layers contribute positively to OOD performance – see detailed results in Sec. 5.5. This intuition aligns with the previous knowledge in the literature that different levels of intermediate layers in a model have different characteristics, e.g., early layers are more general, and later layers are more specific (Yosinski et al., 2014). LEVI uses this knowledge to improve overall results, as it enjoys different levels of intermediate outputs.

Remark 5 (Using a Fine-tuned Large Model)**.** *We note that LEVI's pre-trained model itself can also be replaced with an adapted (i.e., fine-tuned) model. When the given large model is suitable to learn the target downstream task (e.g., using ImageNet pre-trained ViT for ImageNet-related tasks), using the adapted model can be helpful to further improve overall performances of LEVI. In such cases, LEVI's training efficiency is still better than ensembling multiple large models and comparable with fine-tuning one large model.*

Remark 6 (Compatibility with Efficient Training Approaches)**.** *If we replace LEVI's pre-trained model with an adapted (i.e., fine-tuned) model, we can also utilize efficient training methods for large models (Hu et al., 2022; Chen et al., 2020b) like LoRA, which further enhance the efficiency of LEVI – see results in Sec. 5.3.*

5. Experiments

We perform extensive experiments to evaluate LEVI in various scenarios. All experiments are repeated with three random seeds. We use TensorFlow (Abadi et al., 2015) with JAX (Bradbury et al., 2018) and Flax (Heek et al., 2023). More detailed settings (e.g., hyperparameters) are in Sec. D.

Datasets & OOD Scenarios. We consider various OOD scenarios in two modalities: language-based recommendation tasks and computer vision tasks. For language-based recommendation, we use MovieLens (Harper & Konstan, 2015) and Amazon Review (Ni et al., 2019) – see an additional experiment for another natural language processing task in Sec. E.3. For vision, we use Diabetic Retinopathy (Medical) (Emma Dugas, 2015) and ImageNet-Variants (Wang et al., 2019; Hendrycks et al., 2021a;b; Recht et al., 2019). All datasets are from TensorFlow Datasets (TFD). Pre-processing details are in Sec. D.1.

[MovieLens] Contains movie rating data from online movie website, where each data sample has 12 attributes (e.g., user id, movie title, and movie genre). We utilize user-id, user-age, user-occupation, user-zipcode, movie-title, and movie-id as input features and rating as the label attribute, where the rating range is $[1, 5]$. For the OOD scenario, we consider **genre (subpopulation) shifts**: the ID data contains movies of top-5 popular genres (action, comedy, drama, romance, thriller), and the OOD data contains other 12 genres (e.g., animation, sci-fi) that have at least 200 data points.

[Amazon Review] Contains product rating data from the Amazon.com website, where each data sample has 15 attributes (e.g., customer id, product category, product title). We utilize customer-id, product-title, and product-id as input features and rating as the label attribute, where the rating range is $[1, 5]$. For the OOD scenarios, we consider **time shifts** and **product (subpopulation) shifts**: the ID data contains the first 4 years' (oldest) ratings of books, and the OOD data contains the most recent year's ratings of books and other products (e.g., watch, toy, sports, jewelry).

[Diabetic Retinopathy (Medical)] Contains human retina images, where the label attribute is the severity of diabetic retinopathy in the range $[0, 4]$. For the OOD scenario, we consider **quality shifts**: the ID data and OOD data contain high-resolution and low-resolution images, respectively.

[ImageNet-Variants] Contains different styles (e.g., sketch, adversarial) of ImageNet datasets with 1,000 label classes. For the OOD scenario, we consider **domain shifts**: the ID dataset is ImageNet-Sketch, and the OOD datasets are ImageNet-A, ImageNet-R, and ImageNet-V2.

Models. We use two large pre-trained models: T5x (Raffel et al., 2020; Roberts et al., 2022) and ImageNet-21k pre-trained ViT (Dosovitskiy et al., 2021) for language-based recommendation and computer vision tasks, respectively. We note that LEVI can also gracefully work together with other types of model backbones, as LEVI is a model-agnostic approach for improving OOD generalization.

In LEVI, we use a small randomly-initialized model and adapting layers together with the pre-trained model, as explained in Fig. 3. For the small model, we use a two-layer multi-layer perceptron (MLP) model with input embedding layers for recommendation tasks and a four-layer convolutional neural network (CNN) model for computer vision tasks. Each adapting layer is composed of an MLP with one hidden layer. Details on the model configurations (e.g., number of neurons in hidden layers) can be found in Sec. D.2.

Baselines. We compare LEVI with three types of baselines: 1) standard training baselines for pre-trained models, 2) state-of-the-art fine-tuning generalization baselines, and 3) parameter-efficient fine-tuning baselines.

For standard training, we consider four baselines: full fine-tuning (FT), light-tuning of half of the parameters (i.e., half of transformers; HT), light-tuning of the last linear layer (i.e., linear probing; LP), and training-from-scratch (FS). The full fine-tuning baseline updates all pre-trained model parameters, while the light-tuning baselines update them partially. These light-tuning baselines are also known as robust fine-tuning methods for OODs (Kumar et al., 2022a). The training-from-scratch baseline first randomly initializes all parameters then trains on the fine-tuning data.

Table 1. Performances on the MovieLens and Amazon Review datasets. All algorithms are evaluated on separate ID and OOD datasets using the root-mean-square error (RMSE $=\sqrt{\sum_i (\mathrm{y}_i - \hat{\mathrm{y}}_i)^2/m}$), a standard metric for recommendation systems, where lower is better.

	Method	MovieLens		Amazon Review		
		ID	OOD (12 genres)	ID	OOD (time)	OOD (7 products)
Standard Training Baselines	T5x Fine-tuning (FT)	1.175$_{\pm 0.021}$	1.268$_{\pm 0.036}$	1.386$_{\pm 0.024}$	1.543$_{\pm 0.014}$	2.029$_{\pm 0.154}$
	T5x Light-tuning: Half of transformers (HT)	1.230$_{\pm 0.006}$	1.297$_{\pm 0.005}$	1.436$_{\pm 0.035}$	1.529$_{\pm 0.014}$	1.862$_{\pm 0.289}$
	T5x Light-tuning: Linear probing (LP)	1.226$_{\pm 0.014}$	1.218$_{\pm 0.054}$	1.453$_{\pm 0.065}$	1.529$_{\pm 0.045}$	1.759$_{\pm 0.130}$
	T5x From scratch (FS)	1.175$_{\pm 0.021}$	1.291$_{\pm 0.019}$	1.410$_{\pm 0.064}$	1.532$_{\pm 0.014}$	1.494$_{\pm 0.010}$
Fine-tuning Generalization Baselines	LP→FT (Kumar et al., 2022b)	1.163$_{\pm 0.014}$	1.357$_{\pm 0.009}$	1.407$_{\pm 0.003}$	1.503$_{\pm 0.015}$	1.669$_{\pm 0.025}$
	FT+RobustModel (Kumar et al., 2022a)	1.072$_{\pm 0.002}$	1.127$_{\pm 0.038}$	1.374$_{\pm 0.016}$	1.498$_{\pm 0.004}$	1.752$_{\pm 0.130}$
	FT+FS	1.058$_{\pm 0.011}$	1.159$_{\pm 0.018}$	1.333$_{\pm 0.019}$	1.504$_{\pm 0.006}$	1.521$_{\pm 0.029}$
	FT+ZeroShot (Wortsman et al., 2022b)	1.177$_{\pm 0.010}$	1.253$_{\pm 0.038}$	1.336$_{\pm 0.015}$	1.517$_{\pm 0.017}$	2.515$_{\pm 0.175}$
	LEVI	**0.932**$_{\pm \mathbf{0.005}}$	**1.065**$_{\pm \mathbf{0.018}}$	**1.095**$_{\pm \mathbf{0.003}}$	**1.310**$_{\pm \mathbf{0.006}}$	**1.298**$_{\pm \mathbf{0.006}}$

For fine-tuning generalization, we consider four state-of-the-art baselines for large pre-trained models: LP→FT (Kumar et al., 2022b), FT+RobustModel (Kumar et al., 2022a), FT+ZeroShot (Wortsman et al., 2022b), and FT+FS. LP→FT is a two-step baseline that first performs linear probing then full fine-tuning to mitigate the pre-trained feature distortion originated from the randomly initialized head. FT+RobustModel, FT+ZeroShot, and FT+FS are the ensemble baselines. FT+RobustModel ensembles the calibrated outputs of a fine-tuned model and a robustness-aware trained model, which are assumed to achieve good ID and OOD performances, respectively. Here we use linear probing (LP) as the robustness-aware model, as in the original paper (Kumar et al., 2022b). FT+ZeroShot ensembles the parameters of a fine-tuned model and a pre-trained (zero-shot) model to preserve general features in the pre-trained model. FT+FS ensembles the outputs of a fine-tuned model and a trained-from-scratch model to reduce the impacts of problematic pre-trained features via the trained-from-scratch model.

For parameter efficient fine-tuning, we utilize LoRA (Hu et al., 2022), a state-of-the-art method that replaces Transformer layers with rank decomposition matrices to reduce the number of training parameters. We use LoRA in Sec. 5.3 to further improve LEVI's training efficiency.

5.1. Performances on IDs and OODs

Recommendation. Table 1 shows the in-distribution (ID) and out-of-distribution (OOD) performances on the MovieLens and Amazon Review datasets. Here we consider the following OOD scenarios: genre, time, and product shifts. The full results for genre and product shifts are in Sec. E.1.

For the standard training baselines, full fine-tuned models indeed show worse OOD performances than at least one of all the light-tuned (e.g., linear-probed) and trained-from-scratch models, as expected in Sec. 3.1. Notably, for product shifts in Amazon Review (last column of Table 1), the fine-tuned model performs far worse than all the light-tuned and trained-from-scratch models – full results are in Table 11.

Table 2. Performances on the Medical and ImageNet datasets using standard accuracy, where higher is better. We mark the best and second best results with **bold** and underline, respectively.

Method	Medical		ImageNet	
	ID Acc	OOD Acc	ID Acc	OOD Acc
ViT Fine-tuning (FT)	78.58$_{\pm 0.16}$	74.20$_{\pm 0.15}$	83.06$_{\pm 0.05}$	21.62$_{\pm 0.48}$
ViT Light-tuning: HT	77.04$_{\pm 0.31}$	73.63$_{\pm 0.04}$	81.52$_{\pm 0.17}$	28.85$_{\pm 0.30}$
ViT Light-tuning: LP	74.76$_{\pm 0.08}$	71.21$_{\pm 0.38}$	66.48$_{\pm 0.11}$	31.49$_{\pm 0.07}$
LP→FT	**78.83**$_{\pm \mathbf{0.02}}$	73.81$_{\pm 0.03}$	81.39$_{\pm 0.14}$	28.26$_{\pm 0.16}$
FT+RobustModel	78.16$_{\pm 0.01}$	74.16$_{\pm 0.07}$	**84.22**$_{\pm \mathbf{0.15}}$	30.30$_{\pm 0.26}$
FT+ZeroShot	78.35$_{\pm 0.46}$	74.36$_{\pm 0.10}$	82.70$_{\pm 0.13}$	32.10$_{\pm 0.16}$
LEVI	78.60$_{\pm 0.04}$	**75.15**$_{\pm \mathbf{0.05}}$	83.33$_{\pm 0.03}$	**33.62**$_{\pm \mathbf{0.12}}$

The state-of-the-art fine-tuning generalization baselines mostly improve the OOD performances of the fine-tuned model. Among these baselines, we observe that the heavy ensemble approaches FT+RobustModel and FT+FS show promising performances. Also, we note that the FT+ZeroShot baseline, which has a core assumption that the pre-trained (zero-shot) features can give generally good features to the target task, sometimes largely fails to improve the OOD performances in our recommendation tasks (e.g., last column of Table 1). We suspect that the pre-trained (zero-shot) language features are not sufficient for this downstream task and thus may even harm the results.

In contrast, LEVI greatly improves both ID and OOD results in all scenarios by combining two complementing models. This result shows that LEVI effectively suppresses the problematic features in both the fine-tuning data and pre-trained model, while preserving useful features for the task.

Image Classification. Table 2 shows the ID and OOD performances on the Medical and ImageNet-variants datasets with quality shifts and domain shifts, respectively. The full results for ImageNet OOD shifts are in Table 12. We note that as ViT models are known to be hard to train on small or mid-sized training data using random weight initialization (Steiner et al., 2022), we do not use the training-from-scratch baseline in ViT experiments. Interestingly, in

Table 3. Number of parameters and FLOPs of baselines and LEVI on T5x-small. The FLOPs of T5x is obtained from Akbari et al. (2022). See full results including all baselines and ViT in Sec. E.5.

Method	Params (Train.)	Params (Infer.)	FLOPs
Fine-tuning	60M	60M	33G
LP→FT (Kumar et al., 2022b)	60M+∼5K	60M	33G
Baseline ensembles of two models	120M	120M	66G
Baseline ensembles of one model	60M	60M	33G
LEVI with pre-trained model	∼2M	60M+∼2M	33G+∼4M
LEVI with fine-tuned model	60M+∼2M	60M+∼2M	33G+∼4M

the medical dataset, the robust training baselines including light-tuning and LP→FT are not helpful in improving generalization. The major assumption of these approaches is that the pre-trained features have "good-enough" features for supporting downstream tasks, but the ImageNet pre-trained ViT may not have enough features to support the medical image-specific features. On the other hand, in the ImageNet classification, as the ImageNet pre-trained ViT can be considered to already have reasonable features to support ImageNet-variants OOD datasets, the previous robust training baselines indeed improve the generalization of the fine-tuned model. In both datasets, LEVI can achieve the best OOD accuracies among all baselines while having ID accuracies comparable to that of the fine-tuned model.

5.2. Efficiency Comparison

We compare the efficiency of algorithms in Table 3 varying 1) the number of model parameters and 2) floating point operations (FLOPs), which are widely used to estimate required memories and computational costs. When evaluating LEVI, we either use a pre-trained model or its fine-tuned version. When LEVI uses the original pre-trained model, the number of training parameters is significantly lower than all state-of-the-art baselines, and the number of inference parameters and FLOPs are comparable to those of a single large model. LEVI using a fine-tuned model shows comparable results with single model-based baselines in all three metrics. Notably, compared to the heavy ensembles (i.e., FT+RobustModel, FT+FS), which consistently show good performances among the baselines in Tables 1 and 2, LEVI performs training and inference much faster while also achieving better OOD generalization.

5.3. Compatibility with Efficient Fine-Tuning Methods

We evaluate how LEVI using a fine-tuned model performs with LoRA (Hu et al., 2022) to improve its training efficiency. In Table 4, LEVI still improves the ID and OOD results of the LoRA-tuned models, indicating that LEVI can be gracefully merged with existing efficient fine-tuning methods. We provide more discussion on possible extensions of LEVI to further improve its efficiency via other efficient ensemble techniques (e.g., BatchEnsemble) in Sec. F.

Table 4. LoRA results on the Medical and ImageNet datasets.

	Medical		ImageNet	
Method	ID Acc	OOD Acc	ID Acc	OOD Acc
LoRA-tuned ViT	75.23±0.01	73.06±0.14	**79.71±0.09**	22.08±0.17
LEVI w. LoRA-tuned ViT	**76.69±0.17**	**73.57±0.06**	78.78±0.21	**25.64±0.25**

Table 5. Ablation study on the recommendation tasks, where we consider genre shifts for MovieLens and time shifts for Amazon Review. We note that A1 and A2 are the final prediction ensembles, and A3 is the intermediate layer ensemble.

	MovieLens		Amazon Review	
Method	ID RMSE	OOD RMSE	ID RMSE	OOD RMSE
T5x Fine-tuning	1.175±0.021	1.268±0.036	1.386±0.024	1.543±0.014
A1) T5x+T5x	1.103±0.011	1.178±0.021	1.362±0.013	1.531±0.010
A2) T5x+MLP	1.035±0.009	1.163±0.015	1.272±0.008	1.455±0.019
A3) T5x Intermediates	1.125±0.003	1.142±0.012	1.137±0.001	**1.285±0.007**
A4) Single-head MLP	1.003±0.013	1.158±0.101	1.104±0.005	1.326±0.002
A5) Multi-head MLP	1.001±0.020	1.121±0.034	1.102±0.004	1.319±0.036
LEVI	**0.932±0.005**	**1.065±0.018**	**1.095±0.003**	1.310±0.006

5.4. Ablation Study

We perform an ablation study on LEVI to evaluate the impact of each component. We first compare with simple final prediction ensembles (A1) between two fine-tuned large models and (A2) between one fine-tuned large model and one trained-from-scratch small yet task-specialized model, without using intermediate layers. We also compare with (A3) only ensembling intermediate layers without using the trained-from-scratch model. We then compare with solely using task-specialized models with (A4) single-head and (A5) multi-head – see detailed settings in Sec. D.4.

Table 5 shows the comparison with LEVI, where A1 and A2 show both worse ID and OOD results, A3 shows good results in OOD, but not enough in ID, and A4 and A5 achieve good results in ID but not enough in OOD. We conclude that all components in LEVI (i.e., using a small yet task-specialized model with multiple intermediate layers of a large model) contribute to the overall ID and OOD results.

5.5. Effects of Different Intermediate Layers

We investigate the roles of different intermediate layers used in LEVI. We use one intermediate layer at each time and attach a small MLP classification module to the target intermediate layer for training. Interestingly, Fig. 4 shows that the later and early layers tend to be more useful for ID and OOD performances, respectively. This result is consistent with previous work that early layers are more general (robust), and the later layers are more specific (accurate) (Yosinski et al., 2014). A recent study also observes that using multiple intermediate layers can improve the overall model robustness compared to simple linear probing (Evci et al., 2022). Similarly, this result demonstrates that using both the later and early layers can improve the ID-OOD tradeoffs.

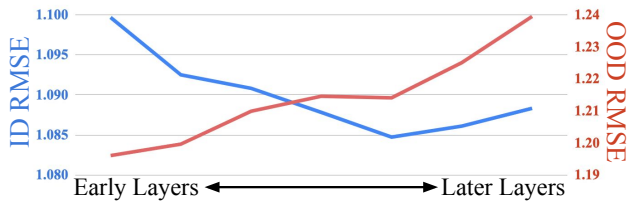


Figure 4. Effects of intermediate layers on ID (blue) and OOD (red) performances, where lower RMSE is better. We report the average results on MovieLens and Amazon Review using T5x.

5.6. Discussion on the distribution gap (domain gap) between pre-training, fine-tuning, and test data

We now discuss how the distribution gap (domain gap) between pre-training, fine-tuning, and test data affects LEVI. Revisiting our results in Tables 1 and 2, we observe different degrees of domain gap. First, Table 1 shows scenarios where the pre-training domain is relatively distant from the downstream fine-tuning and test domains (i.e., using general language pre-trained models in recommendation domains). On the other hand, Table 2 covers cases where the pre-training domain is more similar to the downstream domains (i.e., using image pre-trained models in other image domains). Across both scenarios, LEVI shows clear performance improvements over baseline methods.

In general, we anticipate that LEVI performs especially well when the pre-training and downstream domains differ (e.g., language vs. recommendation domains), as LEVI's advantage comes from breaking the previous assumption that the pre-training domain already has good enough features.

6. Conclusion

We proposed a novel fine-tuning method LEVI, which tightly ensembles a large pre-trained model with a smaller task-specific model for improving OOD generalization of fine-tuning. We first identified that the inherent issues in both pre-trained models and fine-tuning data can jointly hurt the OOD generalization of fine-tuned models. To address these issues, LEVI combines two complementing models to suppress their problems while preserving useful features for downstream tasks, leading to improved OOD generalization, especially when pre-training and downstream distributions are largely different. Experiments on large language and vision models showed that LEVI greatly enhances fine-tuning OOD generalization while not losing ID performances. We believe LEVI reveals the value of using trained-from-scratch models in mitigating the limitations of pre-trained models.

Impact Statement

We believe our work on improving fine-tuning generalization can positively impact society by making AI models being more robust and safe when deployed in real-world applications. Specifically, generalizable AI models would be less prone to failures or unpredictable behaviors when faced with new inputs from unseen distributions, leading to increased reliability and safety. Also, AI models with a strong generalization ability can improve people's lives in areas that require high adaptability across diverse contexts, including healthcare, finance, and recommendation systems.

We do note that as our framework is based on supervised learning that requires training (fine-tuning) data, considering privacy and fairness issues in the training data will become essential. Many recent studies reveal that machine learning models, especially large models applied irresponsibly, could amplify privacy concerns and societal unfairness (Bommasani et al., 2021; Barocas et al., 2023). Thus, one needs to carefully build the training data for LEVI to prevent potential negative impacts, especially when the target applications highly affect society.

References

TensorFlow Datasets, a collection of ready-to-use datasets. https://www.tensorflow.org/datasets.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/.

Akbari, M., Banitalebi-Dehkordi, A., and Zhang, Y. E-LANG: Energy-based joint inferencing of super and swift language models. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2022.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *ArXiv*, 2019.

Barocas, S., Hardt, M., and Narayanan, A. *Fairness and machine learning: Limitations and opportunities*. 2023.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N. S., Chen, A. S., Creel, K. A., Davis, J., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L. E., Goel, K., Goodman, N. D., Grossman, S., Guha, N., Hashimoto,

T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T. F., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M. S., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S. P., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J. F., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y. H., Ruiz, C., Ryan, J., R'e, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K. P., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M. A., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. On the opportunities and risks of foundation models. *ArXiv*, 2021.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Carlucci, F. M., D'Innocente, A., Bucci, S., Caputo, B., and Tommasi, T. Domain generalization by solving jigsaw puzzles. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2229–2238, 2019.

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. S. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *International Conference on Machine Learning (ICML)*, pp. 1691–1703. PMLR, 2020a.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *International Conference on Machine Learning (ICML)*, volume 119, pp. 1691–1703, 2020b.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

Dinh, T., Zeng, Y., Zhang, R., Lin, Z., Gira, M., Rajput, S., Sohn, J.-y., Papailiopoulos, D., and Lee, K. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:11763–11784, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Du, C., Chen, Z., Feng, F., Zhu, L., Gan, T., and Nie, L. Explicit interaction model towards text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 6359–6366, 2019.

D'Innocente, A. and Caputo, B. Domain generalization with domain-specific aggregation modules. In *German Conference on Pattern Recognition (GCPR)*, 2019.

Emma Dugas, Jared, J. W. C. Diabetic retinopathy detection, 2015. URL https://kaggle.com/competitions/diabetic-retinopathy-detection.

Evci, U., Dumoulin, V., Larochelle, H., and Mozer, M. C. Head2toe: Utilizing intermediate representations for better transfer learning. In *International Conference on Machine Learning (ICML)*, pp. 6009–6033, 2022.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning (ICML)*, pp. 1126–1135, 2017.

Fort, S., Ren, J., and Lakshminarayanan, B. Exploring the limits of out-of-distribution detection. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, pp. 1180–1189, 2015.

Geng, S., Liu, S., Fu, Z., Ge, Y., and Zhang, Y. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *ACM Conference on Recommender Systems (RecSys)*, 2022.

Gowal, S., Rebuffi, S.-A., Wiles, O., Stimberg, F., Calian, D. A., and Mann, T. A. Improving robustness using generated data. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:4218–4233, 2021.

Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. Retrieval augmented language model pre-training. In *International conference on machine learning (ICML)*, pp. 3929–3938, 2020.

Harary, S., Schwartz, E., Arbelle, A., Staar, P., Abu-Hussein, S., Amrani, E., Herzig, R., Alfassy, A., Giryes, R., Kuehne, H., et al. Unsupervised domain generalization by learning a bridge across domains. In *IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5280–5290, 2022.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst. (TIIS)*, 5(4), 2015.

He, K., Girshick, R., and Dollár, P. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pp. 4918–4927, 2019.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 9729–9738, 2020.

Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. Flax: A neural network library and ecosystem for JAX, 2023. URL http://github.com/google/flax.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., and Gilmer, J. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *International Conference on Computer Vision (ICCV)*, 2021a.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021b.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International conference on machine learning (ICML)*, pp. 2790–2799, 2019.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Kang, W.-C., Ni, J., Mehta, N., Sathiamoorthy, M., Hong, L., Chi, E., and Cheng, D. Z. Do llms understand user preferences? evaluating llms on user rating prediction. *ArXiv*, 2023.

Khani, F. and Liang, P. Removing spurious features can hurt accuracy and affect groups disproportionately. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pp. 196–205, 2021.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 2661–2671, 2019.

Kotu, V. and Deshpande, B. *Predictive analytics and data mining: concepts and practice with rapidminer*. Morgan Kaufmann, 2014.

Kumar, A., Ma, T., Liang, P., and Raghunathan, A. Calibrated ensembles can mitigate accuracy tradeoffs under distribution shift. In *Uncertainty in Artificial Intelligence (UAI)*, 2022a.

Kumar, A., Raghunathan, A., Jones, R. M., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations (ICLR)*, 2022b.

Kundu, J. N., Venkat, N., Babu, R. V., et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 4544–4553, 2020.

Lee, Y., Chen, A. S., Tajwar, F., Kumar, A., Yao, H., Liang, P., and Finn, C. Surgical fine-tuning improves adaptation to distribution shifts. In *International Conference on Learning Representations (ICLR)*, 2023.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. Learning to generalize: Meta-learning for domain generalization. In *AAAI conference on artificial intelligence (AAAI)*, volume 32, 2018.

Li, R., Jiao, Q., Cao, W., Wong, H.-S., and Wu, S. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 9641–9650, 2020.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Liao, Y., Huang, R., Li, J., Chen, Z., and Li, W. Deep semisupervised domain generalization network for rotary machinery fault diagnosis under variable speed. *IEEE Transactions on Instrumentation and Measurement*, 2020.

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:21464–21475, 2020.

Mahajan, D., Tople, S., and Sharma, A. Domain generalization using causal matching. In *International Conference on Machine Learning (ICML)*, pp. 7313–7324, 2021.

McCoy, T., Pavlick, E., and Linzen, T. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Association for Computational Linguistics (ACL)*, pp. 3428–3448, 2019.

Menon, A. K., Rawat, A. S., Reddi, S. J., and Kumar, S. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations (ICML)*, 2019.

Nagarajan, V., Andreassen, A., and Neyshabur, B. Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations (ICLR)*, 2021.

Ni, J., Li, J., and McAuley, J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019.

Pagliardini, M., Jaggi, M., Fleuret, F., and Karimireddy, S. P. Agree to disagree: Diversity through disagreement for better transferability. In *International Conference on Learning Representations (ICLR)*, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 21(140):1–67, 2020.

Raghunathan, A., Xie, S. M., Yang, F., Duchi, J., and Liang, P. Understanding and mitigating the tradeoff between robustness and accuracy. In *International conference on machine learning (ICML)*, 2020.

Rame, A., Ahuja, K., Zhang, J., Cord, M., Bottou, L., and Lopez-Paz, D. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In *International Conference on Machine Learning (ICML)*, 2023.

Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., and Hsieh, C.-J. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, pp. 5389–5400, 2019.

Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., Dillon, J., and Lakshminarayanan, B. Likelihood ratios for out-of-distribution detection. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning (ICML)*, pp. 4334–4343. PMLR, 2018.

Roberts, A., Chung, H. W., Levskaya, A., Mishra, G., Bradbury, J., Andor, D., Narang, S., Lester, B., Gaffney, C., Mohiuddin, A., Hawthorne, C., Lewkowycz, A., Salcianu, A., van Zee, M., Austin, J., Goodman, S., Soares, L. B., Hu, H., Tsvyashchenko, S., Chowdhery, A., Bastings, J., Bulian, J., Garcia, X., Ni, J., Chen, A., Kenealy, K., Clark, J. H., Lee, S., Garrette, D., Lee-Thorp, J., Raffel, C., Shazeer, N., Ritter, M., Bosma, M., Passos, A., Maitin-Shepard, J., Fiedel, N., Omernick, M., Saeta, B., Sepassi, R., Spiridonov, A., Newlan, J., and Gesmundo, A. Scaling up models and data with `t5x` and `seqio`. *ArXiv*, 2022. URL https://arxiv.org/abs/2203.17189.

Roh, Y., Lee, K., Whang, S., and Suh, C. Fr-train: A mutual information-based approach to fair and robust training. In *International Conference on Machine Learning (ICML)*, pp. 8147–8157, 2020.

Roh, Y., Lee, K., Whang, S. E., and Suh, C. Improving fair training under correlation shifts. In *International Conference on Machine Learning (ICML)*, pp. 29179–29209, 2023.

Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning (ICML)*, pp. 8346–8356, 2020.

Shen, Y. and Sanghavi, S. Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning (ICML)*, pp. 5739–5748, 2019.

Shen, Z., Liu, Z., Li, J., Jiang, Y.-G., Chen, Y., and Xue, X. Dsod: Learning deeply supervised object detectors from scratch. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp. 1919–1927, 2017.

Shen, Z., Liu, J., He, Y., Zhang, X., Xu, R., Yu, H., and Cui, P. Towards out-of-distribution generalization: A survey. *ArXiv*, 2021a.

Shen, Z., Liu, Z., Qin, J., Savvides, M., and Cheng, K.-T. Partial is better than all: Revisiting fine-tuning strategy for few-shot learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9594–9602, 2021b.

Sinha, A., Namkoong, H., and Duchi, J. C. Certifying some distributional robustness with principled adversarial training. In *International conference on learning representations (ICLR)*, 2017.

Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2022.

Steiner, A. P., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., and Beyer, L. How to train your vit? data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research (TMLR)*, 2022. ISSN 2835-8856.

Tian, J., He, Z., Dai, X., Ma, C.-Y., Liu, Y.-C., and Kira, Z. Trainable projected gradient method for robust fine-tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7836–7845, 2023.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 7167–7176, 2017.

Vyas, A., Jammalamadaka, N., Zhu, X., Das, D., Kaul, B., and Willke, T. L. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 550–564, 2018.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 10506–10518, 2019.

Wang, M. and Deng, W. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

Wang, Q., Fink, O., Van Gool, L., and Dai, D. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7201–7211, 2022.

Wen, Y., Tran, D., and Ba, J. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Winkens, J., Bunel, R., Roy, A. G., Stanforth, R., Natarajan, V., Ledsam, J. R., MacWilliams, P., Kohli, P., Karthikesalingam, A., Kohl, S., et al. Contrastive training for improved out-of-distribution detection. *ArXiv*, 2020.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning (ICML)*, pp. 23965–23998, 2022a.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al. Robust fine-tuning of zero-shot models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7959–7971, 2022b.

Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems (NeurIPS)*, 33:6256–6268, 2020.

Xie, S. M., Kumar, A., Jones, R., Khani, F., Ma, T., and Liang, P. In-n-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. In *International conference on learning representations (ICLR)*, 2021.

Xue, Y., Payani, A., Yang, Y., and Mirzasoleiman, B. Eliminating spurious correlations from pre-trained models via data mixing. *ArXiv*, 2023.

Yang, J., Zhou, K., Li, Y., and Liu, Z. Generalized out-of-distribution detection: A survey. *ArXiv*, 2021.

Yi, M., Hou, L., Sun, J., Shang, L., Jiang, X., Liu, Q., and Ma, Z. Improved ood generalization via adversarial training and pretraing. In *International Conference on Machine Learning (ICML)*, pp. 11987–11997, 2021.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? *Advances in neural information processing systems (NIPS)*, 2014.

Zhang, C. and Ma, Y. *Ensemble machine learning: methods and applications*. Springer, 2012.

Zhang, X., Zhou, L., Xu, R., Cui, P., Shen, Z., and Liu, H. Towards unsupervised domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4910–4920, 2022.

Zoph, B., Ghiasi, G., Lin, T.-Y., Cui, Y., Liu, H., Cubuk, E. D., and Le, Q. Rethinking pre-training and self-training. *Advances in neural information processing systems (NeurIPS)*, 33:3833–3845, 2020.

A. More Related Work

Continuing from Sec. 2, we discuss more related work on traditional studies of out-of-distribution (OOD) generalization and other related fields (e.g., model robustness for noisy or adversarial data, domain adaptation, and OOD detection).

OOD Generalization in the Traditional Literature Among the broad model generalization issues, making the model more robust to various OOD scenarios becomes indispensable for AI deployment (Shen et al., 2021a). The main approaches for OOD generalization can be categorized into: 1) unsupervised representation learning that finds a better representation for diverse distributions (Mahajan et al., 2021; Zhang et al., 2022; Harary et al., 2022; Liao et al., 2020), 2) supervised model training that modifies model architectures or training processes to prevent the model from losing the generalization (Finn et al., 2017; Li et al., 2018; D'Innocente & Caputo, 2019; Carlucci et al., 2019; Raghunathan et al., 2020; Xie et al., 2021), and 3) specialized optimization designing that uses robustness-aware objectives to ensure OOD performances (e.g., distributionally robust optimization and invariant risk minimization) (Sinha et al., 2017; Arjovsky et al., 2019).

Among the above approaches, the supervised model training approaches are the most relevant category to ours, as they design new training processes by using given labeled training data. For example, a traditional approach is to utilize meta learning (Finn et al., 2017; Li et al., 2018), where a model trained on a variety of tasks adapts to the new tasks using a small number of training samples. Alternatively, adversarial training-based approaches (Roh et al., 2020; Yi et al., 2021) train a model with perturbed input data or additional discriminator to make the model robust to OOD data. Also, several self-training-based approaches (Raghunathan et al., 2020; Xie et al., 2021) show promising improvements in generalization by leveraging additional unlabeled OOD data. However, many traditional studies do not consider the challenges in large models (e.g., inherent problems in the pre-trained features, scalability) or assume to access pre-training or unlabeled test data. In comparison, our work improves generalization in the fine-tuning pre-trained model paradigm, where 1) the given models may already have inherent issues, 2) the training and inference efficiency becomes more important, and 3) additional information on the pre-training or deployment (OOD) data is usually unavailable.

Other Related Work Although not our immediate focus, there are other noteworthy directions, including 1) model robustness for noisy or adversarial data, which aims to maintain model accuracy even when the input data is corrupted (Song et al., 2022; Gowal et al., 2021; Shen & Sanghavi, 2019; Carmon et al., 2019; Menon et al., 2019; Ren et al., 2018), 2) domain adaptation, which focuses on enabling a model trained on one domain to perform effectively on different yet related domains (Wang & Deng, 2018; Wang et al., 2022; 2021; Li et al., 2020; Kundu et al., 2020; Tzeng et al., 2017; Ganin & Lempitsky, 2015), and 3) OOD detection, which concentrates on identifying data that are not from the training distribution (Yang et al., 2021; Fort et al., 2021; Liu et al., 2020; Winkens et al., 2020; Ren et al., 2019; Vyas et al., 2018). In comparison, we focus on training generalizable models in naturally occurring OODs, but do not explicitly consider adversarial scenarios or OOD detection. Extending LEVI to support these directions can be an interesting future work.

Finally, we discuss other related studies that use relevant ideas with some parts of LEVI's architecture. First, Li & Liang (2021) and Houlsby et al. (2019) use adapting layers (adaptors) in the model architecture for achieving parameter-efficient fine-tuning, where the adaptors are used to improve full fine-tuning performances with fewer parameters. In contrast, we utilize the adapting layers to merge complementing information from two very different models for OOD generalization. Also, Lee et al. (2023) and Shen et al. (2021b) use partial fine-tuning strategies in the OOD setting, but they require small amounts of labeled new domain data (i.e., the OOD test data). In comparison, LEVI does not assume any prior knowledge of OOD data. In addition, Zoph et al. (2020), He et al. (2019), and Shen et al. (2017) show that solely using the trained-from-scratch models can perform better than using fine-tuned pre-training models in object detection and segmentation. In contrast, we jointly use both pre-trained and trained-from-scratch models to utilize their strengths while mitigating their own inherent problems.

B. Synthetic Example

Continuing from Sec. 3.1, we provide a synthetic example illustrating that fine-tuned models can have a worse generalization ability compared to pre-trained and trained-from-scratch models.

Let a target task consists of five features $[x_1, x_2, x_3, x_4, x_5]$, where x_1 and x_5 are spurious features, and x_2, x_3, and x_4 are transferable features. In the following scenario, x_1 and x_5 are originated from pre-trained features and fine-tuning data, respectively. Here are the settings:

- We consider a linear model $\mathrm{y} = \boldsymbol{w}\mathrm{x}^T$. Here, the optimal model weights $\boldsymbol{w}_{\text{true}}$ are $[0, 1, 1, 1, 0]$, which do not rely on the spurious features x_1 and x_5 while only using transferable features.

- *[Fine-tuning data]* We have three fine-tuning data points $(\mathrm{x}^{(1)}, \mathrm{y}^{(1)}) = ([0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}], 1)$, $(\mathrm{x}^{(2)}, \mathrm{y}^{(2)}) = ([0, 0, -\frac{1}{2}, -\frac{1}{2}, 0], -1)$, and $(\mathrm{x}^{(3)}, \mathrm{y}^{(3)}) = ([0, 0, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}], 1)$, which are affected by the spurious feature x_5.

- *[Pre-trained model]* Here, we assume a pre-trained model with weights $\boldsymbol{w}_{\text{pretrain}} = [1, 1, 1, 0, 0]$, which utilizes the spurious feature x_1.

- *[Fine-tuned model]* Here, when we fine-tune the given pre-trained model with the above two data points, the fine-tuned model weights become $\boldsymbol{w}_{\text{finetune}} = [1, 1, 1, 1, 1]$.

- *[Trained-from-scratch model]* On the other hand, when we train a model from scratch, the model weights can be $\boldsymbol{w}_{\text{train-from-scratch}} = [0, 0, 1, 1, 1]$.

As a result, we have three models $\boldsymbol{w}_{\text{pretrain}} = [1, 1, 1, 0, 0]$, $\boldsymbol{w}_{\text{train-from-scratch}} = [0, 0, 1, 1, 1]$, and $\boldsymbol{w}_{\text{finetune}} = [1, 1, 1, 1, 1]$.

Table 6 shows four test data points, where there is no correlation between the label y and spurious features x_1 and x_5. We note that the optimal model with $\boldsymbol{w}_{\text{true}} = [0, 1, 1, 1, 0]$ can predict all test samples correctly with zero errors. When we apply the three models $\boldsymbol{w}_{\text{pretrain}}$, $\boldsymbol{w}_{\text{train-from-scratch}}$, and $\boldsymbol{w}_{\text{finetune}}$ to these test data points, we get the predictions $\hat{\mathrm{y}}$ as in the fourth column in Table 6. As the fine-tuned model $\boldsymbol{w}_{\text{finetune}}$ uses both spurious features x_1 and x_5 while $\boldsymbol{w}_{\text{pretrain}}$ and $\boldsymbol{w}_{\text{train-from-scratch}}$ use either x_1 or x_5, the errors of the fine-tuned model are worse than the other two models' errors (last column in Table 6). This observation is consistent with our previous discussions, including Corollary 3 and the results in Figures 1 & 2.

Table 6. Test data points, model predictions, and errors.

x_{test}	y_{test}	Model	$\hat{\mathrm{y}}$	L1 loss (i.e, $\lvert\mathrm{y}_{\text{test}} - \hat{\mathrm{y}}\rvert$)
$[1, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 1]$	1	$\boldsymbol{w}_{\text{pretrain}}$	5/3	2/3
		$\boldsymbol{w}_{\text{train-from-scratch}}$	5/3	2/3
		$\boldsymbol{w}_{\text{finetune}}$	3	2
$[1, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, 1]$	-1	$\boldsymbol{w}_{\text{pretrain}}$	1/3	4/3
		$\boldsymbol{w}_{\text{train-from-scratch}}$	1/3	4/3
		$\boldsymbol{w}_{\text{finetune}}$	1	2
$[-1, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, -1]$	1	$\boldsymbol{w}_{\text{pretrain}}$	-1/3	4/3
		$\boldsymbol{w}_{\text{train-from-scratch}}$	-1/3	4/3
		$\boldsymbol{w}_{\text{finetune}}$	-1	2
$[-1, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, -1]$	-1	$\boldsymbol{w}_{\text{pretrain}}$	-5/3	2/3
		$\boldsymbol{w}_{\text{train-from-scratch}}$	-5/3	2/3
		$\boldsymbol{w}_{\text{finetune}}$	-3	2

C. Weighted Sum of Loss Functions

Continuing from Sec. 4, we discuss a possible extension of the loss function in our framework.

In Sec. 4, we introduce LEVI's loss function as follows:

$$\min_{\boldsymbol{w}} \frac{1}{m} \cdot \frac{1}{n} \sum_{i}^{m} \sum_{j}^{n} \ell(\mathrm{y}_i, \hat{\mathrm{y}}_i^{(j)}),$$

where m and n are the numbers of training samples and adapting layers, respectively, and $\hat{\mathrm{y}}_i^{(j)}$ is the predicted label from the adapting layer j for the input data i – see the architecture of LEVI in Figure 3, which illustrates each component in the loss function. We update LEVI by using the above loss function, which allows us to tightly ensemble the information from both pre-trained and trained-from-scratch (i.e., randomly initialized then trained) models in the adapting layers.

We basically consider the equal weights on all $\hat{\mathrm{y}}^{(j)}$, but one can change it as a weighted sum of $\hat{\mathrm{y}}^{(j)}$ as follows:

$$\min_{\boldsymbol{w}} \frac{1}{m} \cdot \frac{1}{n} \sum_{i}^{m} \sum_{j}^{n} w^{(j)} \ell(\mathrm{y}_i, \hat{\mathrm{y}}_i^{(j)})$$
$$\text{s.t.} \sum_{j}^{n} w^{(j)} = 1, \;\; w^{(j)} \geq 0,$$

where $w^{(j)}$ is the weight for the loss of each adapting layer j (i.e., $\ell(\mathrm{y}, \hat{\mathrm{y}}^{(j)})$). Here, $w^{(j)}$ can be a hyperparameter to tune the importance between different intermediate layers. For example, when we aim to focus on more specific (general) features, we can give more weight to the later (early) layers.

D. Experimental Settings

Continuing from Sec. 5, we provide more details on experimental settings. In all experiments, we use Dragonfish TPU (i.e., TPUv3) and Jellyfish TPU (i.e., TPUv2) with 2x2 topology for T5x and ViT experiments, respectively. Also, as we discussed in Sec. 5, we use TensorFlow (Abadi et al., 2015) with JAX (Bradbury et al., 2018) and Flax (Heek et al., 2023).

D.1. Datasets and Pre-processings

We consider four datasets: MovieLens (Harper & Konstan, 2015), Amazon Review (Ni et al., 2019), Diabetic Retinopathy (Medical) (Emma Dugas, 2015), and ImageNet-Variants (Wang et al., 2019; Hendrycks et al., 2021a;b; Recht et al., 2019). In this paper, we use MovieLens and Amazon Review for language-based recommendation tasks and Medical and ImageNet-Variants for computer vision tasks. All datasets are from the TensorFlow Datasets library (TFD).

Tables 7 and 8 show the data examples of the MovieLens and Amazon Review datasets.

- MovieLens (Harper & Konstan, 2015) contains movie rating data from online movie website, and we utilize user-id, user-age, user-occupation, user-zipcode, movie-title, and movie-id as input features and rating as the label attribute, where the rating range is $[1, 5]$. We use a stable version of MovieLens that contains 100,000 ratings from 943 users on 1,682 movies. Each user gives ratings on at least 20 movies. For the OOD scenario, we consider *genre (subpopulation) shifts*, where the ID data contains movies of top-5 popular genres (action, comedy, drama, romance, thriller), and the OOD data contains other 12 genres (e.g., animation, sci-fi) that have at least 200 data points. We construct the ID and OOD datasets to be mutually exclusive.
- Amazon Review (Ni et al., 2019) contains product rating data from the Amazon.com website, and we utilize customer-id, product-title, and product-id as input features and rating as the label attribute, where the rating range is $[1, 5]$. For the OOD scenarios, we consider *time shifts* and *product (subpopulation) shifts*. Here, the ID data contains the first 4 years' (oldest) ratings of books, and the OOD data contains the most recent year's ratings of books (i.e., time shifts) and other products (i.e., product shifts), including watch, toy, sports, music, jewelry, furniture, and baby.
- When we serve these tabular-based data into the language model, we use the method proposed in Dinh et al. (2022), which concatenates all attribute values into one sentence. For example, the first row in Table 7 can be converted into an input sentence *"When the user id is 138, the user occupation is doctor, the user zipcode is 53211, the movie title is One Flew Over the Cuckoo's Nest (1975), and the movie id is 357, what can be the user rating on this movie?:"*. Also, when we serve the data into the small MLP model of LEVI, we pre-process the data so that it can be used as the input for the embedding layers. Here, we follow the pre-processing method used in Geng et al. (2022).

Figures 5 & 6 and Figures 7 & 8 show the data examples of the Diabetic Retinopathy (Medical) and ImageNet-Variants datasets, respectively.

- Diabetic Retinopathy (Medical) (Emma Dugas, 2015) contains human retina images, where the label attribute is the severity of diabetic retinopathy in the range $[0, 4]$. For the OOD scenario, we consider *quality shifts*: the ID data (Figure 5) and OOD data (Figure 6) show different image styles and resolutions.
- ImageNet-Variants (Wang et al., 2019; Hendrycks et al., 2021a;b; Recht et al., 2019) contain different styles (e.g., sketch, adversarial) of ImageNet datasets with 1,000 label classes. For the OOD scenario, we consider *domain shifts*:

Table 7. MovieLens data examples.

movie genres	user id	user occupation	user zipcode	movie title	movie id	rating
Drama	138	doctor	53211	One Flew Over the Cuckoo's Nest (1975)	357	4
Comedy, Romance	92	entertainment	80525	Strictly Ballroom (1992)	709	2
Comedy	301	student	55439	Very Brady Sequel, A (1996)	412	4
Crime, Drama	60	healthcare	06472	Pulp Fiction (1994)	56	4
Horror, Thriller	197	technician	75094	Scream 2 (1997)	895	3
Drama, Thriller	601	artist	99687	Crash (1996)	325	4
Animation, Children, ...	710	student	92020	Aladdin (1992)	95	3
Action, Crime, Romance	833	writer	90019	True Romance (1993)	92	2
Comedy	916	engineer	N2L5N	Bob Roberts (1992)	425	5
Adventure, Sci-Fi, ...	920	administrator	02215	Starship Troopers (1997)	271	2

Table 8. Amazon Review data examples. We do not use the review body and title, which strongly indicate the ratings.

product category	review date	customer id	product title	product id	rating
Books	2000-08-10	51389465	Admission Of Love (Arabesque)	1583141642	5
Books	2005-10-04	23641112	Solve Your Child's Sleep Problems	0671620991	4
Watches	2013-08-20	45902750	Timex Men's Expedition Metal Field Watch	B004VRD6FY	3
Toys	2014-07-16	28742093	Ticket To Ride - Europe	B000809OAO	4
Sports	2015-08-29	4005801	Chipolo Bluetooth Item Tracker (Blue)	B00L177Z3Q	1
Music	2009-04-27	10801536	A Mi Edad	B001P0XNJ4	5
Jewelry	2015-02-05	18490528	Sterling Silver Twisted Love Knot Stud Earrings	B00CRMQ8OG	4
Furniture	2013-03-03	28467583	Rust Orange Full Sized with Arms Convertible Sofa	B0070ZERCA	3
Baby	2014-10-15	20757152	Skip Hop Baby Duo Signature Diaper Bag	B00J4J2AQK	5

the ID dataset is ImageNet-Sketch (Figure 7), and the OOD datasets are ImageNet-A, ImageNet-R, and ImageNet-V2 (Figure 8). ImageNet-Sketch consists of about 50,000 black and white sketch images, where each ImageNet class has 50 images. All these datasets share the original ImageNet classes. ImageNet-Sketch (Wang et al., 2019) consists of about 50,000 black and white sketch images, where each class contains 50 images. ImageNet-A (Hendrycks et al., 2021b) consists of adversarial images that are wrongly classified by ResNet-50 models. ImageNet-R (Hendrycks et al., 2021a) consists of rendition images, which contain art, cartoons, graffiti, tattoos, toys, video games, and other renditions. ImageNet-V2 (Recht et al., 2019) consists of new test data for ImageNet, which are collected a decade after the original ImageNet dataset. Previous works (Kumar et al., 2022b; Wortsman et al., 2022b) used an ImageNet pre-trained model, fine-tuned it with a smaller version of ImageNet, and tested it on other ImageNet variants (e.g., ImageNet-A and ImageNet-R). Compared to this setting, we focus on a more challenging scenario where the fine-tuning data is different from the pre-training data; we thus fine-tune the model on ImageNet-Sketch, which has a different image style from the original ImageNet, and test on other variants (ImageNet-A, ImageNet-R, and ImageNet-V2).

- We resize all images into 224×224 pixels.

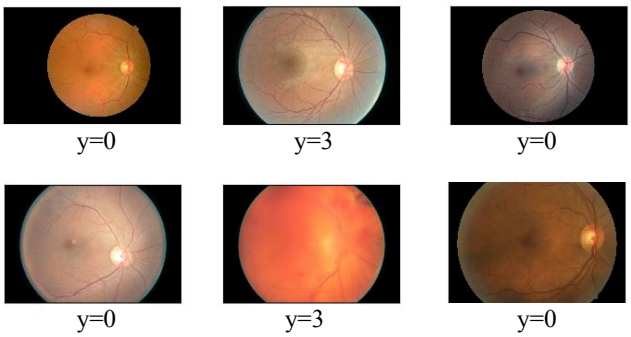


Figure 5. Diabetic Retinopathy (Medical) data examples for the in-distribution. The images are from TensorFlow Datasets (TFD).

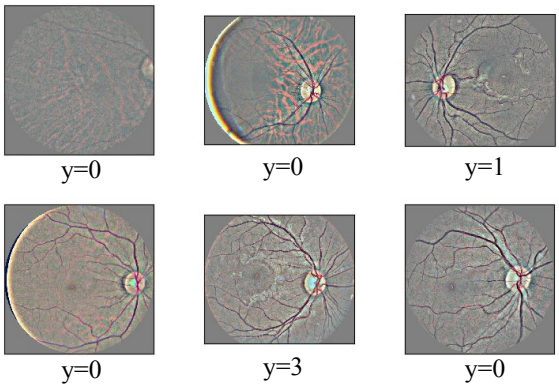


Figure 6. Diabetic Retinopathy (Medical) data examples for the out-of-distribution. The images are from TensorFlow Datasets (TFD).

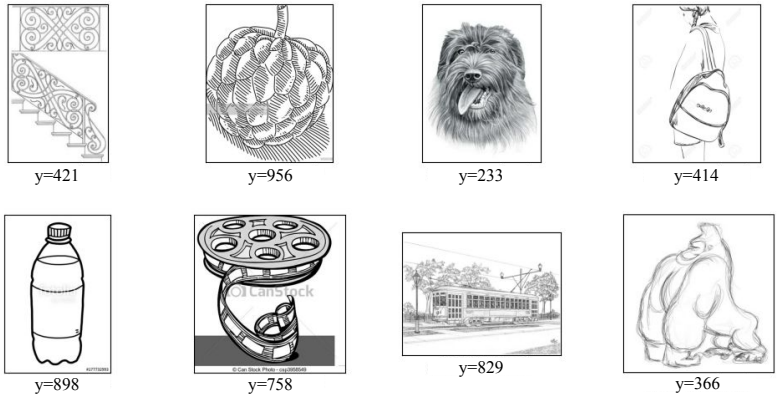


Figure 7. ImageNet-Sketch data examples used for the in-distribution of ImageNet experiments. The images are from TensorFlow Datasets (TFD).

D.2. Models

In our experiments, we use two large pre-trained models: T5x (Raffel et al., 2020; Roberts et al., 2022) and ImageNet-21k pre-trained ViT (Dosovitskiy et al., 2021) for language-based recommendation and computer vision tasks, respectively. Specifically, we use T5x-small and ViT-base architectures, which have 60M and 86M parameters, respectively. We modify the last layers of T5x and ViT to work with each classification task.

LEVI uses a small randomly-initialized model and adapting layers together with the given large model (i.e., T5x and ViT). We note that the small randomly-initialized model in LEVI can be defined to effectively learn the task-specialized features, and we follow the general knowledge from recommendation and computer vision literature.

- For the large model, LEVI can use either an original pre-trained model or an adapted (fine-tuned or light-tuned) model, as discussed in Remark 5. In our main experiments, we perform LEVI with a light-tuned model that updates half of the parameters (i.e., half of the transformers) to improve the overall performance by aligning a given large model with the target task. We also provide the full results of LEVI with pre-trained, light-tuned, and fine-tuned models in Sec. E.4.
- For the small randomly-initizlied model, we use a two-layer multi-layer perceptron (MLP) model with input embedding layers for recommendation tasks and a four-layer convolutional neural network (CNN) model for computer vision tasks. The two-layer MLP model used in the recommendation tasks consist of 512 and 256 neurons in each hidden layer. The CNN model used in the vision tasks consist of four convolution layers ((3×3)-kernel, 32 features), ((3×3)-kernel, 64 features), ((3×3)-kernel, 64 features), and ((3×3)-kernel, 64 features), where a batch normalization, ReLU, and max pooling are followed by each convolution layer. The last layer is flattened to be served to the adapting layers. We note

(a) ImageNet-A (b) ImageNet-R (c) ImageNet-V2

Figure 8. ImageNet-A (a), ImageNet-R (b), and ImageNet-V2 (c) data examples used for out-of-distributions of ImageNet experiments. The images are from TensorFlow Datasets (TFD).

that, in general, as the small model size increases (e.g., it has more layers), LEVI's performance increases, but eventually converges. In our experiments, the small models with good enough performances are much smaller than any large pre-trained models used in the language-based and vision-based tasks.

- Each adapting layer is composed of an MLP with one hidden layer, where the numbers of neurons in the hidden layer are 256 and 1024 for the recommendation and computer vision tasks, respectively.

D.3. Hyperparameters

We provide more details on hyperparameters and settings.

Here are common hyperparameters and settings for all algorithms. We use the Adam optimizer and SGD optimizer for T5x and ViT experiments, respectively. For batch sizes, we use 200 for MovieLens, 100 for Amazon Review, and 512 for all computer vision datasets. For learning rates, we consider a set $\{0.0001, 0.001, 0.01, 0.1\}$ for all algorithms except linear probing. In linear probing, we use a learning rate set with larger values $\{0.001, 0.01, 0.1, 1.0, 10.0\}$, as reported in previous studies (Kumar et al., 2022b; He et al., 2020; Chen et al., 2020a). For each algorithm, we choose the best hyperparameters from the above candidate sets to achieve the best performance in the in-distribution validation set, while not accessing the out-of-distribution datasets.

When the baselines require other hyperparameters (e.g., FT+ZeroShot (Wortsman et al., 2022b) uses a weight parameter to balance the importance between the fine-tuned model and the zero-shot model), we follow the candidate sets used in the original paper and choose the best hyperparameters to achieve the best performance in the in-distribution validation set.

D.4. Settings of Ablation Study

We provide more details on the setting of the ablation study. We consider five ablations (A1–A5), as shown in Table 5. First, we perform a simple prediction ensemble between two fine-tuned large models (A1). We also perform a simple prediction ensemble between a fine-tuned large model and a small trained-from-scratch model (A2). We note that both A1 and A2 do not use intermediate layer information and simply average the final outputs (predictions) of the models. We then perform an ensemble between intermediate layers of a large model without using the trained-from-scratch model (A3). Finally, we compare with trained-from-scratch (task-specialized) models with (A4) single-head and (A5) multi-head without using the large model. Here, A4 is a standard model that produces one prediction per input sample. A5 consists of a shared-bottom layer block followed by multiple final heads that produce predictions, where we average the predictions of the different heads, similar to the ensemble.

Based on the above ablations, we investigate the importance of each component in LEVI. For example, A1 and A3 show the impact of using the small yet task-specialized trained-from-scratch model in LEVI. Also, A2, A4, and A5 show the importance of using the large model. Finally, A1 and A2 show the benefits of using the intermediate layer outputs in the large model. As a result, Table 5 show that all components in LEVI (i.e., using a small yet task-specialized model with multiple intermediate layers of a large model) contribute to the overall ID and OOD performances.

E. Additional Experiments

Continuing from Sec. 5, we provide additional experimental results.

E.1. More Results on Recommendation Tasks

Continuing from Sec. 5.1, we provide full OOD performances on the MovieLens genre shifts in Tables 9 & 10 and Amazon Review product shifts in Table 11. We compare LEVI with 1) standard training baselines (e.g., full fine-tuning, light-tuning, and training-from-scratch) and 2) state-of-the-art fine-tuning generalization baselines.

The results are consistent with those in Table 1. For example, in many cases, full fine-tuned models show worse OOD results than either the light-tuned or trained-from-scratch models. This phenomenon is especially notable in Table 11, where the fine-tuned model performs far worse than all the light-tuned and trained-from-scratch models. Also, the fine-tuning generalization baselines (e.g., LP→FT, FT+RobustModel) mostly improve the OOD performances of the fine-tuned model. However, the FT+ZeroShot baseline sometimes largely fails to improve the OOD performances, as in Table 11. As we discussed in Sec. 5.1, we suspect that the pre-trained (zero-shot) language features are not sufficient for this downstream task and thus may even harm the results. In comparison, LEVI further improves OOD results in all types of shifts.

Table 9. OOD performances on the MovieLens dataset, where the type of OOD is genre shifts (part 1).

Method	Adventure	Sci-Fi	War	Crime	Children	Horror
T5x Fine-tuning (FT)	1.251±0.036	1.314±0.046	1.153±0.033	1.341±0.005	1.262±0.045	1.355±0.050
T5x Light-tuning: Half of transformers (HT)	1.283±0.018	1.352±0.022	1.276±0.019	1.405±0.089	1.298±0.028	1.403±0.008
T5x Light-tuning: Linear probing (LP)	1.233±0.005	1.231±0.099	1.166±0.224	1.402±0.118	1.195±0.032	1.398±0.123
T5x From scratch (FS)	1.350±0.022	1.296±0.018	1.246±0.032	1.367±0.060	1.279±0.031	1.399±0.046
LP→FT (Kumar et al., 2022b)	1.335±0.008	1.352±0.013	1.294±0.071	1.340±0.002	1.368±0.030	1.378±0.004
FT+RobustModel (Kumar et al., 2022a)	1.122±0.013	1.175±0.045	1.066±0.068	1.203±0.026	1.103±0.013	1.257±0.063
FT+FS	1.156±0.018	1.196±0.024	1.076±0.002	1.235±0.010	1.142±0.032	1.259±0.042
FT+ZeroShot (Wortsman et al., 2022b)	1.227±0.032	1.338±0.036	1.134±0.020	1.343±0.003	1.239±0.038	1.363±0.029
LEVI	**1.036±0.006**	**1.138±0.010**	**1.059±0.069**	**1.178±0.051**	**1.047±0.017**	**1.105±0.019**

Table 10. OOD performances on the MovieLens dataset, where the type of OOD is genre shifts (part 2).

Method	Mystery	Musical	Animation	Western	Film-Noir	Documentary
T5x Fine-tuning (FT)	1.268±0.023	1.265±0.060	1.262±0.049	1.216±0.068	1.125±0.082	1.404±0.045
T5x Light-tuning: Half of transformers (HT)	1.245±0.038	1.315±0.024	1.327±0.023	1.194±0.013	1.122±0.014	1.467±0.031
T5x Light-tuning: Linear probing (LP)	1.213±0.122	1.158±0.025	1.166±0.015	1.100±0.101	1.062±0.273	1.288±0.062
T5x From scratch (FS)	1.303±0.043	1.259±0.031	1.262±0.031	1.187±0.038	1.160±0.025	1.388±0.044
LP→FT (Kumar et al., 2022b)	1.428±0.083	1.356±0.049	1.396±0.034	1.327±0.039	1.260±0.004	1.455±0.029
FT+RobustModel (Kumar et al., 2022a)	1.125±0.084	1.097±0.028	1.092±0.020	1.048±0.088	1.017±0.127	**1.221±0.073**
FT+FS	1.167±0.033	1.143±0.028	1.139±0.026	1.091±0.044	1.033±0.018	1.277±0.024
FT+ZeroShot (Wortsman et al., 2022b)	1.254±0.016	1.235±0.049	1.246±0.033	1.124±0.096	1.143±0.076	1.396±0.026
LEVI	**0.973±0.013**	**1.077±0.031**	**1.074±0.015**	**0.949±0.029**	**0.878±0.052**	1.267±0.031

Table 11. OOD performances on the Amazon Review dataset, where the type of OOD is product shifts.

Method	Watch	Toy	Sports	Music	Jewelry	Furniture	Baby
T5x Fine-tuning (FT)	2.016±0.131	2.015±0.181	2.024±0.165	1.986±0.164	2.029±0.158	2.075±0.125	2.057±0.151
T5x Light-tuning: Half of transformers (HT)	1.925±0.291	1.875±0.304	1.814±0.287	1.645±0.295	1.966±0.239	1.916±0.304	1.892±0.303
T5x Light-tuning: Linear probing (LP)	1.776±0.120	1.782±0.129	1.752±0.135	1.668±0.165	1.763±0.123	1.792±0.110	1.782±0.125
T5x From scratch (FS)	1.564±0.004	1.505±0.007	1.471±0.012	1.241±0.029	1.553±0.004	1.593±0.003	1.532±0.012
LP→FT (Kumar et al., 2022b)	1.692±0.023	1.693±0.032	1.657±0.026	1.551±0.010	1.676±0.022	1.716±0.034	1.696±0.029
FT+RobustModel (Kumar et al., 2022a)	1.757±0.111	1.754±0.141	1.745±0.139	1.686±0.162	1.759±0.123	1.789±0.108	1.776±0.129
FT+FS	1.561±0.015	1.526±0.031	1.506±0.031	1.341±0.047	1.562±0.024	1.604±0.022	1.550±0.031
FT+ZeroShot (Wortsman et al., 2022b)	2.509±0.149	2.467±0.169	2.527±0.144	2.647±0.150	2.481±0.176	2.478±0.218	2.498±0.221
LEVI	**1.346±0.003**	**1.303±0.003**	**1.275±0.003**	**1.118±0.013**	**1.348±0.007**	**1.366±0.010**	**1.327±0.002**

E.2. More Results on Vision Tasks

Continuing from Sec. 5.1, we provide full OOD performances on the ImageNet-variant datasets in Table 12. We use the same baselines in the previous section except the trained-from-scratch baseline, as ViT models are known to be hard to train on small or mid-sized training data using random weight initialization (Steiner et al., 2022).

As we observed in Sec. 5.1, the existing fine-tuning generalization baselines improve OOD performances, as the ImageNet pre-trained ViT has reasonable features to support ImageNet-variants datasets. Although these baselines show more effective results in the ImageNet scenario compared to the language-based recommendation tasks, LEVI still achieves the best or second-best OOD accuracies among all baselines while having ID accuracies comparable to that of the fine-tuned model – see the ID results in Table 2.

Table 12. OOD performances on the ImageNet-variant datasets: ImageNet-A, ImageNet-R, and ImageNet-V2.

Method	OOD Acc: A	OOD Acc: R	OOD Acc: V2
ViT Fine-tuning (FT)	1.90±0.22	33.60±0.74	29.37±0.49
ViT Light-tuning: Half of transformers (HT)	4.46±0.26	38.48±0.21	43.62±0.45
ViT Light-tuning: Linear probing (LP)	**8.57±0.04**	35.59±0.05	**50.32±0.12**
LP→FT (Kumar et al., 2022b)	4.43±0.19	36.90±0.15	43.44±0.15
FT+RobustModel (Kumar et al., 2022a)	5.22±0.08	39.98±0.54	45.71±0.15
FT+ZeroShot (Wortsman et al., 2022b)	5.21±0.03	43.35±0.16	47.75±0.32
LEVI	5.25±0.02	**45.94±0.19**	49.68±0.14

E.3. Results on Another NLP Task

Continuing from Sec. 5, we perform an additional experiment for sentiment classification, a different type of natural language processing (NLP) tasks, and observe consistent performance improvements when using LEVI.

In this experiment, we revisit the Amazon Review dataset (Ni et al., 2019), which can be used for the sentiment analysis of customers using their review texts (Xie et al., 2020; Du et al., 2019). We note that while "review texts" have not been used as an input feature in our recommendation experiments to follow the common setting of the recommendation task, we now use the text information as the main input feature for classifying customer sentiments. For the baselines, we compare four standard training baselines (FT, HT, LP, and FS) and a state-of-the-art fine-tuning generalization baseline FT+RobustModel (Kumar et al., 2022a), which shows the best performances among the baselines in Table 1.

As a result, Table 13 shows that LEVI outperforms all the baselines in terms of both ID and OOD performances in the sentiment classification task. This result indicates that LEVI can support more general types of NLP tasks.

Table 13. Performances on the Amazon Review dataset for sentiment classification.

Method	ID RMSE	OOD RMSE
T5x Fine-tuning (FT)	0.686	0.802
T5x Light-tuning: Half of transformers (HT)	0.743	0.868
T5x Light-tuning: Linear probing (LP)	1.343	1.487
T5x From scratch (FS)	0.900	1.089
FT+RobustModel (Kumar et al., 2022a)	0.624	0.727
LEVI	**0.587**	**0.691**

E.4. LEVI with Different Large Models

Continuing from Secs. 5.1 and D.2, we provide the results of LEVI with pre-trained, light-tuned, and fine-tuned models. As LEVI's large model can be one of original pre-trained or downstream task adapted (i.e., light-tuned or fine-tuned) models, we compare the performances of LEVI with different large models.

Table 14 shows the ID and OOD performances of LEVI on the MovieLens and Amazon Review datasets. The ID performances are better when using the fully fine-tuned large models in LEVI compared to using the pre-trained and light-tuned models. In comparison, the OOD performances are better when using the pre-trained and light-tuned models. This result shows the relationship between the ID-OOD tradeoff and adapting large models to the fine-tuning (ID) data. LEVI can enjoy better ID performances when using a large model more adapted to the fine-tuning (ID) data, while achieving better OOD performances with an original pre-trained model, which is less affected by the problematic features in the fine-tuning data.

Table 14. Performances of LEVI on the MovieLens and Amazon Review datasets with different large models.

	MovieLens		Amazon Review		
Method	ID	OOD (12 genres)	ID	OOD (time)	OOD (7 products)
LEVI with pre-trained T5x	$0.944_{\pm 0.004}$	$1.104_{\pm 0.003}$	$1.109_{\pm 0.003}$	$\mathbf{1.296}_{\pm \mathbf{0.011}}$	$\mathbf{1.275}_{\pm \mathbf{0.006}}$
LEVI with light-tuned T5x	$0.932_{\pm 0.005}$	$\mathbf{1.065}_{\pm \mathbf{0.018}}$	$1.095_{\pm 0.003}$	$1.310_{\pm 0.006}$	$1.298_{\pm 0.006}$
LEVI with fully fine-tuned T5x	$\mathbf{0.927}_{\pm \mathbf{0.004}}$	$1.085_{\pm 0.014}$	$\mathbf{1.093}_{\pm \mathbf{0.001}}$	$1.318_{\pm 0.004}$	$1.524_{\pm 0.110}$

E.5. More Results on Efficiency Comparison

Continuing from Sec. 5.2, we provide the full results of efficiency comparison between LEVI and baselines. We use 1) the number of model parameters and 2) floating point operations (FLOPs). Tables 15 and 16 show the comparisons for T5x and ViT, respectively. As we observed in Sec. 5.2, when LEVI uses a pre-trained model, the number of training parameters is much lower than all state-of-the-art baselines, and the other two metrics are comparable to those of using a single large model. When LEVI uses a fine-tuned model, it shows comparable results with single model-based baselines, including FT+ZeroShot, in all three metrics. Compared to the heavy ensembles with two large models (i.e., FT+RobustModel, FT+FS), which show good performances among the baselines in various tasks, LEVI is much efficient while also achieving better OOD generalization.

Table 15. Number of parameters and FLOPs of baselines and LEVI with T5x. The FLOPs of the default T5x model are obtained from the previous study (Akbari et al., 2022).

Method	Params (Train.)	Params (Infer.)	FLOPs
Fine-tuning (FT)	60M	60M	33G
Light-tuning: Half of transformers (HT)	30M	60M	33G
Light-tuning: Linear probing (LP)	∼5K	60M	33G
From scratch (FS)	60M	60M	33G
LP→FT (Kumar et al., 2022b)	60M + ∼5K	60M	33G
FT+RobustModel (Kumar et al., 2022a)	120M	120M	66G
FT+FS	120M	120M	66G
FT+ZeroShot (Wortsman et al., 2022b)	60M	60M	33G
LEVI with pre-trained model	∼2M	60M + ∼2M	33G + ∼4M
LEVI with light-tuned model	30M + ∼2M	60M + ∼2M	33G + ∼4M
LEVI with fully fine-tuned model	60M + ∼2M	60M + ∼2M	33G + ∼4M

Table 16. Number of parameters and FLOPs of baselines and LEVI with ViT. The FLOPs of the default ViT model are obtained from the previous study (Rao et al., 2021).

Method	Params (Train.)	Params (Infer.)	FLOPs
Fine-tuning (FT)	86M	86M	17.5G
Light-tuning: Half of transformers (HT)	43M	86M	17.5G
Light-tuning: Linear probing (LP)	∼1M	86M	17.5G
LP→FT (Kumar et al., 2022b)	86M + ∼1M	86M	17.5G
FT+RobustModel (Kumar et al., 2022a)	172M	172M	35G
FT+ZeroShot (Wortsman et al., 2022b)	86M	86M	17.5G
LEVI with pre-trained model	∼15M	86M + ∼15M	17.5G + ∼5G
LEVI with light-tuned model	43M + ∼15M	86M + ∼15M	17.5G + ∼5G
LEVI with fully fine-tuned model	86M + ∼15M	86M + ∼15M	17.5G + ∼5G

E.6. More Results on Compatibility with Efficient Fine-Tuning Methods

Continuing from Sec. 5.3, we provide full OOD performances when using LoRA (Hu et al., 2022) on the ImageNet-variant datasets. When LEVI uses a fine-tuned model instead of an original pre-trained model, we can use efficient fine-tuning techniques like LoRA to improve the overall training efficiency. Table 18 shows that LEVI improves the OOD performances of the LoRA-tuned models in all the three ImageNet-variant datasets (ImageNet-A, ImageNet-R, and ImageNet-V2), demonstrating that LEVI can be gracefully merged together with existing efficient fine-tuning methods.

Table 17. OOD performances of LoRA experiments on the ImageNet-variant datasets: ImageNet-A, ImageNet-R, and ImageNet-V2.

Method	OOD Acc: A	OOD Acc: R	OOD Acc: V2
LoRA-tuned ViT	2.50$\pm$0.01	31.21$\pm$0.36	32.53$\pm$0.14
LEVI with LoRA-tuned ViT	**3.76**$\pm$**0.09**	**35.91**$\pm$**0.38**	**37.25**$\pm$**0.27**

E.7. More Results on Compatibility with Efficient Fine-Tuning Methods

Continuing from Sec. 5.3, we provide full OOD performances when using LoRA (Hu et al., 2022) on the ImageNet-variant datasets. When LEVI uses a fine-tuned model instead of an original pre-trained model, we can use efficient fine-tuning techniques like LoRA to improve the overall training efficiency. Table 18 shows that LEVI improves the OOD performances of the LoRA-tuned models in all the three ImageNet-variant datasets (ImageNet-A, ImageNet-R, and ImageNet-V2), demonstrating that LEVI can be gracefully merged together with existing efficient fine-tuning methods.

Table 18. OOD performances of LoRA experiments on the ImageNet-variant datasets: ImageNet-A, ImageNet-R, and ImageNet-V2.

Method	OOD Acc: A	OOD Acc: R	OOD Acc: V2
LoRA-tuned ViT	2.50$\pm$0.01	31.21$\pm$0.36	32.53$\pm$0.14
LEVI with LoRA-tuned ViT	**3.76**$\pm$**0.09**	**35.91**$\pm$**0.38**	**37.25**$\pm$**0.27**

E.8. Illustration on Spurious Feature Mitigation

Continuing from Sec. 5, we demonstrate that LEVI can mitigate the impact of spurious features. For a clear illustration, we perform a synthetic experiment with datasets consisting of two spurious features (s1, s2), one transferable feature (x), and one binary label attribute (y). Specifically, we construct three datasets: 1) a pre-training dataset, where s1 and x are correlated with y; 2) a fine-tuning dataset, where s2 and x are correlated with y; and 3) a test dataset, where only x is correlated with y. All datasets have 2,000 data points.

We compare three models, 1) fine-tuned, 2) trained-from-scratch, and 3) LEVI-based models. 1) We first pre-train a 3-layer neural network on the given pre-training data and then fine-tune it on the fine-tuning data. 2) We also prepare a trained-from-scratch model only using the fine-tuning data. 3) Finally, we train a LEVI model that uses complementing views from the pre-trained model and the trained-from-scratch model.

When applying the three models on the test data (only has the transferable feature), we have the following result: the fine-tuned, trained-from-scratch, and LEVI models achieve accuracies (the higher the better) of 68.9, 69.5, and 75.5, respectively. This result indicates that LEVI is clearly less affected by the spurious correlations by s1 and s2 and uses the information of transferable feature x, compared to the fine-tuned and trained-from-scratch models.

F. Possible Extension via Other Efficient Ensemble Techniques

Continuing from Sec. 5.3, we provide more discussion on a possible extension of LEVI to further improve its efficiency via other efficient ensemble techniques. Here, we focus on a widely used efficient ensemble technique called BatchEnsemble (Wen et al., 2020), which is designed to reduce the computational and memory costs of typical heavy ensembles of multiple models, while achieving similar performances with the heavy ensembles. The key idea of BatchEnsemble is to use one shared weight matrix with multiple rank-one matrices, where each rank-one matrix is multiplied by the shared matrix to recover each member (model) of the original ensemble – please refer to the details in Wen et al. (2020).

To further improve the efficiency of LEVI, we can consider applying the idea of BatchEnsemble, especially to the adapting layers in LEVI. When downstream tasks and models become much more complex and large, LEVI may require larger-sized adapting layers compared to those used in our current experiments. In such cases, one possible way to reduce the computational and memory costs can be to set each adapting layer with a single-rank matrix, as in BatchEnsemble. Although LEVI is already very efficient compared to existing heavy ensembles, extending our work by studying the compatibility with other efficient ensemble methods will be a promising future direction.