Supplementary Material

We provide details omitted in the main paper.

- Appendix A: related work (cf. subsection 2.3 and section 5 of the main paper).
- Appendix B: additional benchmark details (cf. subsection 2.2 of the main paper).
- Appendix C: additional training details (cf. section 4 of the main paper).
- Appendix D: additional results and analyses (cf. section 4 of the main paper).
- Appendix E: additional discussions (cf. section 5 of the main paper).

To keep the same reference numbers as in the main paper, we use plain text for those newly added references in the supplementary material.

A Related Work

We review related work on other transfer learning paradigms. We briefly describe their settings and distinguish their differences from our proposed holistic transfer (HT) problem.

A.1 Domain Adaptation

Domain adaptation (DA) is the most iconical machine learning setting to tackle the domain-shift problem (Liu et al., 2021; Chen et al., 2022; Shen et al., 2022; Rangwani et al., 2022; Gandelsman et al., 2022; Yang et al., 2023). With the common objective of transferring source-domain knowledge to target domains, various settings have been proposed to incorporate different constraints and assumptions. The assumption can be the degrees of overlap between the source and the target label sets (Busto et al., 2017; Cao et al., 2018; You et al., 2019; Saito et al., 2020; Yang et al., 2022; Jang et al., 2022). To relax the constraint of accessing source data, source-free DA can solely rely on the target data for adaptation (Ding et al., 2022; Kundu et al., 2022; Chhabra et al., 2023). Despite the abundant variations, DA settings all share one common assumption: the target distributions in training and testing are matched, making our HT fundamentally different from them. In our HT, we can encounter target test classes that are unseen in the target training set but seen in the source domain. Therefore, HT requires a distinct ability that can generalize the style shifts learned on the target seen classes to other unseen classes.

A.2 Out-of-domain Generalization

Although fine-tuning a pre-trained model often leads to impressive accuracy for downstream tasks, recent studies have revealed that it may compromise the out-of-domain (OOD) robustness of the model [20, 2, 37, 21]. Several robust fine-tuning methods are thus proposed to balance the trade-off between in-domain downstream accuracy and OOD generalization (Raghunathan et al., 2020; Xie et al., 2020; Tian et al., 2023). LP-FT [21] proposed to learn a classifier with frozen features before end-to-end fine-tuning to avoid feature distortion. Some other approaches relied on ensembles with pre-trained models to increase the robustness (Wortsman et al., 2022; Ilharco et al., 2022). However, the main focus of these studies remains on preserving the robustness to different input styles for classes seen in the target training set. This is significantly different from HT. Our HT problem aims to generalize the styles for classes unseen in the target training set.

A.3 Continal Learning

The goal of continual learning (CL) is to sequentially adapt to multiple tasks without catastrophically forgetting the previously learned ones [16, 26, 24]. To achieve this goal, existing studies have proposed to exploit a replay buffer for storing old data (Rebuffi et al., 2017; Chaudhry et al., 2019; Wu et al., 2022; Tiwari et al., 2022; Yoon et al., 2022; Luo et al., 2023; Zhu et al., 2023; Zhou et al., 2023), or to constrain the fine-tuning with old models (Yoon et al., 2017; Dhar et al., 2019; Ahn et al., 2019; Douillard et al., 2022; Wang et al., 2022). Unlike HT, CL still assumes all the encountered training distributions, which could be many, are aligned with their corresponding test distributions. Although reducing forgetting can be the first step for HT to maintain unseen class accuracy, we

Table 10: A summary of the dataset statistics for our HT benchmark.

Datasets	Source domains	Target domain	#Classes	#Seen classes	#Target training	#Target test
Office-Home	Art	Clipart	65	30	1,471	1,330
		Product			1,265	1,361
		Real			1,413	1,335
	Real	Art	65	30	857	750
		Clipart			1,493	1,330
		Product			1,459	1,361
FEMNIST	40 writers	10 new writers	62	Vary by data collection bias	Vary by data collection bias	Vary by data collection bias
iWildCam	53 camera trap locations	21 new camera trap locations	181	Vary by data collection bias	Vary by data collection bias	Vary by data collection bias
VTAB	CLIP	Caltech101	102	51	1,371	6,084
		CIFAR100	100	50	22,513	10,000
		DTD	47	23	920	1,880
		EuroSAT	10	5	8,424	5,400
		Flowers102	102	51	510	6,149
		Pets	37	18	1,445	3,669
		Resisc45	45	22	9,159	6,300
		SVHN	10	5	28,197	26,032
		SUN397	397	198	37,542	21,750
iNaturalist (Fungi)	CLIP	Fungi	12	6	30	60

argue that this is insufficient in HT due to the source-target domain mismatch. Adapting the features for unseen classes to the target domain remains a key challenge for HT. Moreover, HT can also be potentially compatible with CL to consider learning on a non-iid data stream.

A.4 Zero-shot Learning

Zero-shot learning tackles the setting where training and test classes are completely disjoint (Xian et al., 2017; Chen et al., 2021; Xu et al., 2022; Pourpanah et al., 2022). As no training data are available for test classes, the main challenge resides in learning source features that can generalize to unseen semantic meanings. To achieve this, auxiliary information (e.g., texts or attributes) is usually needed to describe the test classes and connect them back to the training classes (Xu et al., 2020; Naeem et al., 2021; Chen et al., 2022; Li et al., 2022). In HT, we assume the missing classes in target domains are already seen in the source domain. We make this assumption to simplify the problem so that HT can focus on generalizing the domain shifts to unseen classes. However, we argue that HT is compatible with zero-shot learning to make the setting more flexible.

B Additional Benchmark Details

To support the study of the HT problem, we create a benchmark that covers extensive scenarios across both experimental and realistic public datasets. We provide details about these datasets.

B.1 Office-Home

Setup. We consider the standard domain adaptation setting but with some missing classes in the target training sets. We use the popular Office-Home dataset consisting of 65 categories from 4 domains (Art, Clipart, Real, and Product). In our benchmark, we use Art and Real as source domains; each source domain is then transferred to each of the three remaining target domains individually, resulting in six source-target pairs. For each source-target pair, we use all the data in the source domain to train a source model. Then, for each target domain, we randomly split the data of each class into training and test sets with a ratio of 7:3. We randomly sample 30 seen classes and combine the training data of these seen classes to create the target training set. Finally, the target test set consists of the test images of all 65 classes in the target domain. A summary of the statistics can be found in Table 10.

Evaluation. We follow the standard evaluation metric in the Office-Home dataset to compute the overall accuracy for each source-target pair. Besides, we explicitly compute the accuracy on the unseen-class data to evaluate the transferring performance of the unseen classes. The average accuracy over all the source-target pairs is also reported.

B.2 FEMNIST

Setup. The FEMNIST dataset contains 62-class hand-written characters from many writers with different writing styles. As we can only collect a limited-size data set for each writer, each writer's data only cover a subset of the 62-class characters, resulting in the need for HT. We randomly sample 40 writers whose data combined can cover all 62 classes and use their data to train a source model. Then, we randomly sample 10 new writers. Each new writer's data is divided into training and test sets in a ratio of 7:3. Note that each client may not have enough images per class, which creates a realistic scenario of personalization with limited samples, which results in a mismatch of the class distributions between training and test sets. The dataset statistics are summarized in Table 10.

Evaluation. We report the overall accuracy averaged over all the 10 new writers. To evaluate the trade-off between seen and unseen classes, we also report the averaged accuracy on the seen and unseen classes, respectively. As this dataset has no oracle training set for each new writer, we report the seen accuracy computed by chopping out unseen classes in the classifier to evaluate the quality of the adapted features.

B.3 iWildCam

Setup. We consider a realistic scenario of HT, where we initially have abundant camera traps installed across many geo-locations (source domains) and now need to transfer to a new camera trap location (target domain). In the new location, we can only use the data collected within a fixed amount of time in the beginning (e.g., the first month) as our target training set. As it is impossible for all the animal species to appear in the first month, the target training data can bias toward some classes that show up. This is a natural data collection bias caused by time.

We start from the iWildCam dataset in the WILDS [17] benchmark. As we mainly focus on animal species classification, we remove the "empty" class for simplicity and thus obtain a total of 181 classes. For each camera trap location, we sort the images by their timestamps and group images into sequences if the difference in their timestamps is smaller than 30 minutes, to avoid information leaks. We randomly sample 53 camera trap locations whose images cover all 181 classes and use all their data to train a source model. For each of the remaining locations, we randomly sample training and test sets based on a ratio of 7:3. We only keep locations with more than 500 images in both the training and test sets, thereby resulting in 21 new locations for adaptation. For each new location, we form the target training set by sorting the training images by time and only using the first 25% of them. A summary of the dataset statistics is given in Table 10.

Evaluation. We report the overall accuracy averaged over all 21 new locations. To evaluate the trade-off between seen and unseen classes, we also report the averaged accuracy on the seen and unseen classes, respectively. As this dataset has no oracle training set for each new location, we report the seen accuracy computed by chopping out unseen classes in the classifier to evaluate the quality of the adapted features.

B.4 VTAB

Setup. We consider another practical use of HT by going beyond domain adaptation and fine-tuning the zero-shot CLIP [30] for distribution shifts at the task levels. We use the VTAB [39] benchmark that includes various image classification tasks. To enable zero-shot predictions, we only use the tasks that provide text names for classes, thereby resulting in 9 tasks: Caltech101, CIFAR100, DTD, EuroSAT, Flowers102, Pets, Resisc45, SVHN, and SUN397. We use the standard training and test sets provided by the VTAB benchmark. Then, we randomly sample half of the classes as seen and the remaining as unseen. The target training set only includes the training images of the seen classes, while the target test set contains all the test images. A summary of the statistics of this dataset is shown in Table 10.

Evaluation. Following the standard evaluation in VTAB, we report the overall accuracy for each of the 9 tasks. Besides, we also compute the accuracy on the unseen-class data to evaluate the transferring performance of unseen classes. Finally, the average accuracy across all 9 tasks is also reported.

B.5 iNaturalist (2021 Version, Fungi)

Setup. To demonstrate the impact of visually similar classes in HT, we carefully pick 6 pairs of fungi classes from the iNaturalist dataset, thus resulting in a total of 12 classes. Each pair of fungi classes corresponds to 2 species of visually similar fungi; one is non-toxic, while the other one is toxic. We use the zero-shot CLIP model with the fungi names as our source model. Then, the training images from the 6 non-toxic fungi classes form the target training set. The target test set consists of all the test images from all 12 classes. A summary of the dataset statistics and some examples are shown in Table 10.

Evaluation. We report the seen accuracy on the target test set to evaluate the adaptation performance. As wrongly predicting toxic fungi as non-toxic ones can result in severe outcomes, we also report the false negative rate, which is computed as the percentage of the images of toxic fungi being predicted as non-toxic fungi classes.

C Additional Training Details

We provide the training details for our results reported in section 4.

For the Office-Home dataset, we initialize a ResNet-50 with ImageNet pre-trained weights. Then, we train it on the source domain for 20 epochs using the SGD optimizer with a learning rate 1e-3, momentum 0.9, weight decay 5e-4, and batch size 64. For all methods that adapt to the target domains, we fine-tune the source model for 20 epochs using the SGD optimizer with a learning rate 1e-4, momentum 0.9, weight decay 5e-4, and batch size 64. For our suggested HT methods, we set the hyper-parameters $\mathcal{L}_{\text{distill}} = 10$ and $\mathcal{L}_{\text{rank}} = 100$.

For the FEMNIST dataset, we train a LeNet from scratch on the data of the 40 source writers for 100 epochs using the SGD optimizer with a learning rate 1e-2, momentum 0.9, weight decay 5e-4, and batch size 32. To adapt to each new writer, we fine-tune the source model for 10 epochs using the SGD optimizer with a learning rate 1e-3, momentum 0.9, weight decay 1e-4, and batch size 32. We set the hyper-parameters $\mathcal{L}_{\text{distill}} = 0.1$ and $\mathcal{L}_{\text{rank}} = 10$.

For the iWildCam dataset, we train a ResNet-50, which is initialized with ImageNet pre-trained weights, on the data of source camera trap locations for 50 epochs using the SGD optimizer with a learning rate 3e-5, momentum 0.9, weight decay 0.0, and batch size 16. When adapting to each new location, we fine-tune the source model for 20 epochs using the SGD optimizer with a learning rate 3e-6, momentum 0.9, weight decay 0.0, and batch size 16. We set the hyper-parameters $\mathcal{L}_{\text{distill}} = 50$ and $\mathcal{L}_{\text{rank}} = 200$.

For the VTAB benchmark, we use the class names for each of the 9 tasks to form the zero-shot CLIP models, which are ViT-B/32. We fine-tune the source model on target tasks for 20 epochs using the SGD optimizer with a learning rate 1e-5, momentum 0.9, weight decay 0.0, and batch size 64. We set the hyper-parameters $\mathcal{L}_{\text{distill}} = 1$ and $\mathcal{L}_{\text{rank}} = 5$.

For the iNaturalist Fungi dataset, we use the fungi species names to build a zero-shot CLIP model with a ViT-B/32 architecture. We then fine-tune the source model on the target training set for 5 epochs using the SGD optimizer with a learning rate 5e-5, momentum 0.9, weight decay 0.0, and batch size 5. We set the hyper-parameters $\mathcal{L}_{\text{distill}} = 1$ and $\mathcal{L}_{\text{rank}} = 1$.

D Additinal Results and Analyses

D.1 Variances of the Results in section 4

We provide variances of our results reported in our main paper. We compute the variances across 3 random seeds. Table 11 shows the variances of the test accuracy on Office-Home. The variances of the mean accuracy on FEMNIST and iWildCam are provided in Table 12 and in Table 13, respectively. Finally, Table 14 gives the variances of the test accuracy for each of the 9 tasks in VTAB. These results reveal that the reported accuracy is relatively robust across random seeds.

Table 11: Varainces of domain adaptation 65-way test accuracy on Office-Home with 30 seen and 35 unseen classes (cf. Table 3). We compute the variances over 3 random seeds. Blue: HT methods suggested by us in section 3. Red: methods that significantly improve overall accuracy and successfully maintain unseen accuracy on the source model. ❄: the linear classifier is frozen during training.

Domains: source→target	Ar→Cl		Ar→Pr		Ar→Rw		Rw→Ar		Rw→Cl		Rw→Pr		**Avg.**	
Methods / Acc.	Overall	Unseen	Overall	Unseen	Overall	Unseen	Overall	Unseen	Overall	Unseen	Overall	Unseen	Overall	Unseen
Source	0.64	0.11	1.30	1.35	0.25	0.24	0.82	0.90	0.23	1.40	0.30	0.49	0.05	0.11
Naive Target	0.02	0.01	0.14	0.02	0.27	0.53	0.93	1.19	0.00	0.36	0.35	1.11	0.04	0.09
BN only	1.11	3.11	0.22	0.27	0.41	1.07	1.71	7.58	0.68	2.44	1.43	4.47	0.03	0.17
BN (stats) only	0.63	0.06	1.00	1.63	0.02	0.32	0.08	0.17	0.16	0.31	0.15	0.05	0.14	0.16
BN (stats) only	0.57	0.05	0.87	0.58	0.25	0.18	0.05	0.02	0.21	0.02	0.30	0.37	0.12	0.03
LP-FT	0.14	0.14	0.05	0.04	0.08	0.18	0.16	0.61	0.01	0.31	0.35	0.87	0.02	0.04
SGD (w/ frozen classifier ❄)	0.02	0.52	0.63	1.27	0.75	0.98	0.07	0.31	0.57	0.85	0.09	0.60	0.07	0.18
SGD + $\mathcal{L}_{\text{distill}}$ ❄	0.28	0.09	0.03	0.01	0.70	0.73	0.23	0.94	0.05	0.09	0.15	1.02	0.02	0.01
SGD + $\mathcal{L}_{\text{rank}}$ ❄	1.02	0.78	0.39	0.48	0.88	1.13	0.01	0.51	0.07	0.67	0.09	0.03	0.08	0.05
SWA ❄	0.23	0.73	0.62	1.44	0.33	0.60	0.45	0.31	1.39	3.39	0.13	0.65	0.02	0.15
SWAD ❄	0.43	0.83	0.62	1.79	0.35	0.42	0.45	0.32	1.64	3.71	0.08	0.44	0.03	0.17
LOLSGD❄	0.36	0.20	0.62	1.00	0.20	1.25	0.73	0.61	1.20	1.59	0.17	0.65	0.00	0.11
LOLSGD +$\mathcal{L}_{\text{rank}}$ ❄	0.02	0.05	0.24	0.95	0.06	0.05	0.34	0.22	0.63	1.09	0.16	0.96	0.04	0.14
LOLSGD +$\mathcal{L}_{\text{distill}}$ ❄	0.14	0.56	0.79	0.54	0.25	0.09	0.22	0.17	0.28	1.88	0.07	0.08	0.06	0.01
LOLSGD +$\mathcal{L}_{\text{distill}}$ + $\mathcal{L}_{\text{rank}}$ ❄	0.18	0.06	0.75	1.82	0.00	0.34	0.26	0.65	0.75	0.36	0.01	0.23	0.12	0.09
Oracle	0.05	0.05	0.06	0.16	0.01	0.16	0.17	0.02	0.22	0.05	0.35	0.20	0.02	0.00

Table 12: Varainces of FEMNIST mean accuracy of 10 new writers (cf. Table 5). We compute the variances over 3 random seeds.

Methods	Overall	Seen	Seen (Chopping)	Unseen
Source	0.44	0.12	0.07	5.12
Naive Target	0.49	0.77	0.77	5.69
SGD❄	0.43	0.17	0.11	1.90
SGD + $\mathcal{L}_{\text{rank}}$ ❄	0.55	0.08	0.04	4.05
SGD + $\mathcal{L}_{\text{distill}}$ ❄	0.37	0.19	0.13	1.72
LOLSGD❄	0.48	0.10	0.17	4.73
LOLSGD +$\mathcal{L}_{\text{rank}}$ ❄	0.37	0.10	0.20	2.04
LOLSGD +$\mathcal{L}_{\text{distill}}$ ❄	0.48	0.16	0.37	4.90
LOLSGD +$\mathcal{L}_{\text{distill}}$ +SE❄	0.65	0.26	0.06	6.23

Table 13: Variances of iWildCam mean accuracy of 21 new locations (cf. Table 6). We compute the variances over 3 random seeds.

Methods	Overall	Seen	Seen (Chopping)	Unseen
Source	0.12	1.45	5.29	0.86
Naive Target	1.66	3.85	4.45	0.04
SGD❄	0.10	1.48	1.67	0.01
SGD + $\mathcal{L}_{\text{rank}}$ ❄	0.94	7.73	2.41	2.35
SGD + $\mathcal{L}_{\text{distill}}$ ❄	3.76	1.26	1.32	4.07
LOLSGD❄	0.79	0.37	0.76	1.63
LOLSGD +$\mathcal{L}_{\text{rank}}$ ❄	1.70	7.85	5.03	3.98
LOLSGD +$\mathcal{L}_{\text{distill}}$ ❄	1.33	0.67	1.86	3.36
LOLSGD +$\mathcal{L}_{\text{distill}}$ +SE❄	1.17	1.07	2.58	0.15

D.2 Different Numbers of Images per Seen Class

In the real world, it is unrealistic for end-users to collect data for all classes before adaptation. To further consider a lower data collection cost, we reduce the number of training images per seen class to study its effects. We conduct the experiment on the Office-Home dataset with "Art" as our source domain and "Clipart" as our target domain. Specifically, we randomly sample 10% and 50% of the training images for each seen class and fine-tune the source model for the *same iterations* for fair comparisons. Interestingly, Table 15 shows that naive fine-tuning can obtain higher unseen accuracy, compared to naive fine-tuning with more data. The reason might be that training with more data needs to update the model weights more, making the unseen classes easier to be forgotten. In contrast, applying our suggested HT methods, especially for LOLSGD with our regularization, the unseen classes can be better maintained across different training data sizes.

D.3 Effects of the Source Ensemble Coefficients

In section 4, we apply Source Ensemble (with a mixing coefficient $\alpha = 0.5$) to reclaim some ability of the source model to maintain the unseen accuracy (cf. Table 4). To further understand the trade-off between the source and the fine-tuned target models, we study the effects of the mixing coefficient α by varying it between $[0, 1]$. We conduct our study on Office-Home and report the overall and unseen accuracy averaged over all the source-target domain pairs. As shown in Figure 6, applying either SE or WiSE cannot save the naively fine-tuned target model from being heavily biased to seen classes. On SGD with frozen classifiers, our SE shows a better trade-off than WiSE [37]. Finally, fine-tuning target models with our suggested HT methods can clearly yield the best trade-off.

Table 14: Variances of test accuracy for fine-tuning CLIP ViT-B/32 on VTAB (cf. Table 7). 50% of classes in each task are missing during training. We compute the variances over 3 random seeds.

Overall/Unseen Acc.	Caltech101		CIFAR100		DTD		EuroSAT		Flowers102		Pets		Resisc45		SVHN		SUN397		Avg.	
Methods	All.	Uns.	All.	Uns.	All.	Uns.	All.	Uns.	All.	Uns.	All.	Uns.	All.	Uns.	All.	Uns.	All.	Uns.	All.	Uns.
Source	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Naive Target	0.00	0.00	0.02	0.08	0.17	1.32	0.01	0.02	0.30	0.45	0.05	0.38	0.22	1.45	0.01	0.03	0.01	0.05	0.00	0.01
SGD❄	0.00	0.01	0.02	0.17	0.31	0.33	0.21	0.59	0.16	0.07	0.03	0.15	0.13	1.00	0.03	0.02	0.00	0.01	0.02	0.02
LOLSGD❄	0.01	0.00	0.01	0.03	0.39	1.33	0.16	0.69	0.21	0.12	0.03	0.13	0.01	0.07	2.29	0.14	0.06	0.29	0.04	0.01
LOLSGD +$\mathcal{L}_{\text{distill}}$ ❄	0.00	0.00	0.01	0.04	0.04	0.00	0.13	0.20	0.01	0.02	0.01	0.00	0.01	0.02	1.80	0.01	0.00	0.00	0.01	0.00
LOLSGD +$\mathcal{L}_{\text{rank}}$ ❄	0.00	0.01	0.00	0.02	0.07	0.07	0.17	0.34	0.06	0.02	0.02	0.01	0.01	0.10	1.37	0.36	0.03	0.06	0.01	0.00
LOLSGD +$\mathcal{L}_{\text{rank}}$ +SE❄	0.01	0.02	0.07	0.14	0.12	0.33	0.06	0.05	0.06	0.03	0.00	0.03	0.05	0.09	1.91	0.82	0.01	0.05	0.00	0.02
Oracle	0.29	0.19	0.00	0.01	0.40	0.50	0.01	0.07	0.27	0.35	0.02	0.14	0.02	0.12	0.00	0.00	0.00	0.01	0.02	0.03

Table 15: Different percentages of the target training data for each seen class on Office-Home: Ar → Cl.

% of target training	10%		50%		100%	
Methods/Acc.	Overall	Unseen	Overall	Unseen	Overall	Unseen
Source	47.07	50.29	47.07	50.29	47.07	50.29
Naive Target	41.73	19.74	43.46	9.94	44.96	9.06
SGD❄	51.28	41.96	52.33	28.36	52.11	24.12
SGD + $\mathcal{L}_{\text{rank}}$ ❄	50.45	46.49	56.02	40.20	59.17	39.47
SGD + $\mathcal{L}_{\text{distill}}$ ❄	48.80	41.37	53.91	38.30	56.54	39.18
LOLSGD❄	**52.63**	46.20	54.21	34.94	56.47	35.09
LOLSGD +$\mathcal{L}_{\text{rank}}$ ❄	51.13	48.83	55.86	44.74	58.57	43.86
LOLSGD +$\mathcal{L}_{\text{distill}}$ ❄	51.28	45.91	55.19	44.88	57.44	46.35
LOLSGD +$\mathcal{L}_{\text{distill}}$ + $\mathcal{L}_{\text{rank}}$ ❄	51.05	**50.88**	**58.65**	**52.05**	**60.83**	**51.75**

E Additional Discussions

E.1 Limitations

In this paper, we introduce a novel and practical transfer learning problem, holistic transfer, that emphasizes the generalization to domain shifts for classes unseen in the target domain but seen in the source domain. We establish strong baselines and demonstrate the potential for simultaneously improving both seen and unseen target classes. One potential limitation is that we mainly focus on vision classification tasks. We leave the studies to image segmentation/object detection and natural language processing tasks as our future work. We also plan to explore better approaches for the disentanglement of domain styles and classes and to integrate our approach with other learning paradigms, like test-time adaptation.

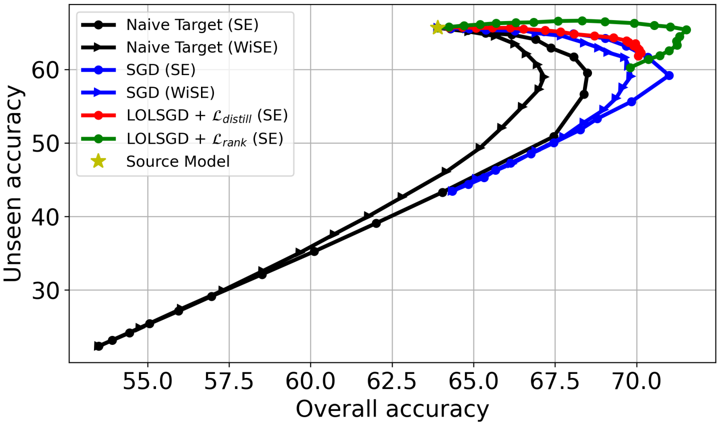

Figure 6: **Effects of Source Ensembles.** Ensemble of the source (the star marker) and the target models (the end point of each line from the source model) with a mixing coefficient $\alpha \in [0, 1]$ on Office-Home.

E.2 Potential Negative Societal Impact

The goal of our work is to introduce and study a practical transfer learning problem, holistic transfer. We provide strong baselines and analyze the problem on publicly available datasets, which are adjusted and split to meet our problem setting. As far as we know, our work does not introduce additional negative societal impacts compared to the standard transfer learning topics, like domain adaptation and out-of-distribution generalization.

E.3 Computation Resources

We conduct our experiments on PyTorch and on NVIDIA V100 GPUs. On the Office-Home dataset, fine-tuning for 1 target domain with all the compared methods and random seeds takes roughly 36 hours on 1 GPU. Similar time consumption also applies to iWildCam and VTAB datasets. On the smaller FEMNIST dataset, it takes roughly 0.5 hours on 1 GPU to get the required results for 1 target domain. The whole experiment on iNaturalist Fungi takes roughly 0.5 on 1 GPU. In total, our experiments take roughly 1.3K GPU hours.