

# Appendix

## A. DETAILED EXPERIMENTAL SETUPS

### A.1. Detailed Setup for Sec. 4.1

**Data Partitioning and Evaluation.** To experiment under CI-FFREEDA condition, we employ the following data partitioning strategy. Initially, we select a source domain and randomly sample instances for each label to create an imbalanced source dataset, representing approximately 60% of the entire dataset. From this subset, 80% is allocated to the training set, while the remaining 20% is designated as the validation set. Subsequently, we select one target domain. For the test set, we allocate an equal number of samples for each label, comprising 20% of the total dataset. The remaining samples are divided using Dirichlet sampling method, commonly employed in non-IID federated learning, to ensure class imbalance and heterogeneous distributions across clients. The samples assigned to each client are stratified to preserve the label distribution, with 80% allocated to the training set and 20% to the validation set.

In many existing domain adaptation studies that do not incorporate federated learning, data partitioning during target adaptation is not conducted, and models are typically evaluated using metrics derived from the training set after a fixed number of epochs. Contrary to their approach, our objective is to obtain a unified global model adapted to the target domain while avoiding overfitting to the training data held by each client. Consequently, we maintain the best global model on the server based on the evaluations conducted by each client using their validation set and measure MAR of the final global model using the test set from the administrator’s perspective. Since the test set is balanced, MAR is equivalent to accuracy.

In the source training phase, three training/validation sets are constructed with different imbalance ratios. For each source imbalance ratio, experiments are conducted three times with different seed values, and the results of these nine runs are averaged. In the target adaptation phase, for each of the three source imbalance ratios, three different target imbalance ratios (representing imbalance distribution among clients) are applied, and the results from these nine runs are averaged.

**Implementation Details.** We adopt mini-batch SGD with momentum 0.9. The batch size is 64 and the learning rate is set to 0.01 for OH and 0.001 for VisDA. The learning rate of backbone during the training of ResNet is set to one-tenth of the default rate. The feature dimension in bottleneck layer is set to 256. In ICPR, RandAugment [1] is used to generate augmented images. The number of augmentations to be given is set to 2 and the upper limit of augmentation magnitude is set to 9. The degree of non-IID-ness of the data distribution, as determined by the Dirichlet sampling, is controlled by the parameter  $\alpha$ . Generally, non-IID-ness is more pronounced when  $\alpha < 1$ , resulting in one label being distributed preferentially to biased clients. In our experiments,  $\alpha$  is set to 0.5.

In experiments using ViT-S and ViT-B, the output of training samples processed by the ViT is stored in the feature bank for training, except during ICPR source training, ICPR target adaptation, and ISFDA target adaptation. Consequently, the data augmentation of random flip and random crop commonly employed in existing methods are not applied during training.

### A.2. Detailed Setup for Sec. 5.1

In the further experiments in Sec. 5.1, three domain dataset in Office-Home; Clipart, Product, Real-World are split into three subset for source-set, target-set, and evaluation-set. In all scenarios, the evaluation-set is balanced. Meanwhile, the source-set is sampled with balanced label distribution in the source balance scenario (*sb*), whereas random label distribution is applied in the source imbalance scenario (*si*). The target-set is further distributed among three clients, maintaining the label distribution in the target balance scenario (*tb*) and applied a Dirichlet distribution in the target imbalance scenario (*ti*). This leads to the following four scenarios: source balance to target balance (*sbtb*), source balance to target imbalance (*sbt*i**), source imbalance to target balance (*sitb*), and source imbalance to target imbalance (*siti*). In Sec. 5.1, only the results of *sbtb* and *siti* are presented. Note that in those experiments, the number of samples used for training in both the source and target domains is approximately half of that in Sec. 4, thus a direct comparison is not possible.

### A.3. Detailed Setup for Sec. 5.2

In the experiments described herein, SHOT is employed as the base algorithm for SFDA. When integrated with FedProx, the regularization term is added to the existing SHOT loss function. The regularization factor is tuned from  $\{1.0, 0.1, 0.01, 0.001\}$ . A value of 0.001 is selected in VisDA with ResNet-101, while 0.1 is selected for the other settings. For FedETF, training is conducted using a dedicated bottleneck component and an ETF classifier. Following the official implementation, we replace the loss function in the self-training term with a balanced softmax loss, where the cross-entropy is corrected using the label distribution. Other implementation details are same as Sec. 4.

## B. COMPLEMENTARY RESULTS

### B.1. Complementary Results of Sec. 4.2

The results of adaptation experiments for all domain patterns in OH are presented in Fig. 5. In each domain-specific panel, different methods are represented by different colors. “Source” refers to the baseline evaluation where the model, trained exclusively on the source domain, is directly applied to the target domain without any adaptation. “Local” represents the average performance across all clients, where each client independently trains its model without federated learning. “Hard”

denotes a basic pseudo-labeling strategy: during local client training, the outputs of the current model are converted into a one-hot format and used as pseudo-labels. Across nearly all methods, the accuracy follows the order: ViT-B (B), ViT-S (S), ResNet-50 (R). The accuracy among all methods is largely competitive, with no method significantly outperforming the others beyond the range of the error bars.

### B.2. Complementary Results of Sec. 5.1

The whole results with TL and DA settings under *sbtb*, *sbt*, *sitb*, and *siti* scenarios are shown in Table 6. The source accuracy is averaged across three domains, three source sampling seeds, and three execution seeds, totaling 27 runs. The target accuracy is averaged across three domain pairs, three source sampling seeds, and three target distribution seeds in TL setting. In DA setting, it is averaged over six source-target pairs, three source sampling seeds, and three target distribution seeds, totaling 27 runs for TL and 54 runs for DA.

**Table 6:** The whole results conducted in Sec.5.1 under transfer learning (TL) and domain adaptation (DA) settings considering domain gaps and label distribution gaps. The Scenario column represents the label distribution of the source (*s*) and target (*t*), balanced (*b*) and imbalanced (*i*). The decline in accuracy after the transfer or adaptation to the target (S2T diff.) is also presented.

	Model	Scenario	Source acc.	Target acc.	S2T diff.
TL	ResNet-50	<i>sbtb</i>	82.6	82.0	-0.6
		<i>sbt</i>	82.6	81.0	-1.6
		<i>sitb</i>	78.7	78.4	-0.3
		<i>siti</i>	78.7	77.6	-1.1
	ViT-S	<i>sbtb</i>	87.4	86.4	-1.0
		<i>sbt</i>	87.4	86.0	-1.4
		<i>sitb</i>	84.7	84.1	-0.6
		<i>siti</i>	84.7	83.4	-1.3
	ViT-B	<i>sbtb</i>	89.9	90.1	+0.2
		<i>sbt</i>	89.9	89.7	-0.2
		<i>sitb</i>	88.8	87.9	-0.9
		<i>siti</i>	88.8	87.2	-1.6
DA	ResNet-50	<i>sbtb</i>	82.6	65.0	-17.6
		<i>sbt</i>	82.6	64.0	-18.6
		<i>sitb</i>	78.7	62.0	-16.7
		<i>siti</i>	78.7	60.8	-17.9
	ViT-S	<i>sbtb</i>	87.4	76.8	-10.6
		<i>sbt</i>	87.4	74.2	-13.2
		<i>sitb</i>	84.7	73.1	-11.6
		<i>siti</i>	84.7	71.4	-13.3
	ViT-B	<i>sbtb</i>	89.9	82.6	-7.3
		<i>sbt</i>	89.9	80.8	-9.1
		<i>sitb</i>	88.8	79.4	-9.4
		<i>siti</i>	88.8	78.1	-10.7

### B.3. Complementary Results of Sec. 5.2

Fig. 6 illustrates the error bars with different seeds for OH dataset, as indicated Table 5. Fig. 7 is the same figure for VisDA.

## C. OTHER MODELS AND BACKBONE FINE-TUNING

In edge environments, such as a single client participating in federated learning, models like ViT-S/14 (21M parameters) and ViT-B/14 (86M), mainly used in this study, may not be sufficiently lightweight for efficient processing. To investigate more resource-efficient alternatives, we include in our evaluation lighter models with available pretrained weights: LightViT-Tiny [2] (9.4M), TinyViT-5M [3] (5.4M), and the image encoder ViT-8M/16 of TinyCLIP [4] (8M). The results are shown for cases where these models are used as frozen backbones, as well as when the pretrained weights are used for initialization and the entire backbone is fine-tuned. Furthermore, the results also include the performance of DINOv2 when the backbone is fine-tuned, as well as the results using the latest models released by Meta: DINOv3 [5], ViT-S/16 (21M), ViT-S+/16 (28M), and ViT-B/16 (86M), used as frozen backbones. For each setting, we conducted a parameter search over learning rates from  $\{0.1, 0.01, 0.001\}$ , weight decay values  $\{0.01, 0.001, 0.0001\}$ , and backbone learning rate ratios during fine-tuning  $\{0.1, 0.01, 0.001, 0.0001\}$ . The best-performing results from these configurations with SHOT and FedAvg are reported.

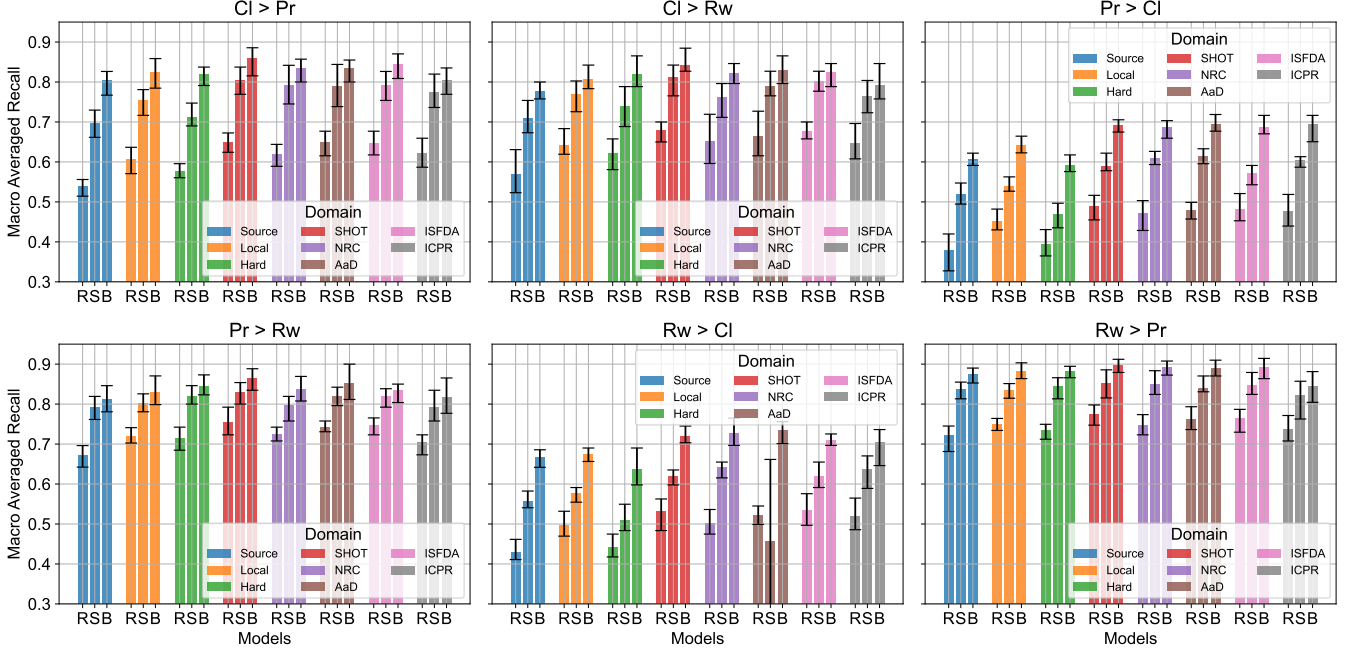
**Source Training Accuracy.** The training results on the source domain show that TinyViT achieves comparable to DINOv2 ViT-S among lightweight models in OH (Fig. 8), while both LightViT and TinyViT demonstrate high accuracy on VisDA (Fig. 9). When fine-tuning lightweight models, the source accuracy remains nearly the same as in the frozen case on OH, whereas higher accuracy is observed in VisDA.

**Target Adaptation Accuracy.** After target adaptation, the frozen TinyViT achieves performance nearly equivalent to DINOv2 ViT-S in OH (Fig. 10) and slightly lower in VisDA (Fig. 11), yet in both cases it outperforms the fine-tuned ResNet. On the other hand, fine-tuning lightweight models results in decreased accuracy, suggesting that more sensitive adjustment may be necessary. Fine-tuning DINOv2 yields accuracy comparable to or slightly better than its frozen counterpart, but considering computational costs, using the frozen backbone proves to be advantageous.

## D. COMPUTATIONAL AND COMMUNICATION COSTS

The proposed method in this study reduces computational resource consumption during backpropagation by freezing the VFM component. Table 7 shows the number of FLOPs and the size of models that must be transferred between the server and the client. FLOPs are computed using the `calcflops`<sup>3</sup> library. Since batch normalization is used, the number of batch is calculated as 2 and the result is halved to obtain the FLOPs per image. Additionally, when fine-tuning, the computational cost of the backward pass is assumed to be twice that of the

<sup>3</sup><https://github.com/MrYxJ/calculate-flops.pytorch>



**Fig. 5:** The whole results of federated target adaptation with OH and ResNet-50 (R), DINOv2 ViT-S (S), and ViT-B (B) conducted in Sec. 4. Panels and colors indicate different source-target domain pairs and SFDA methods, respectively. Each plot shows the average over nine runs, comprised of three different source imbalance ratios and three different target imbalance ratio. Error bars represent the maximum and minimum values from these nine runs.

forward pass. Model sizes are based on the actual saved size using the standard method of PyTorch.

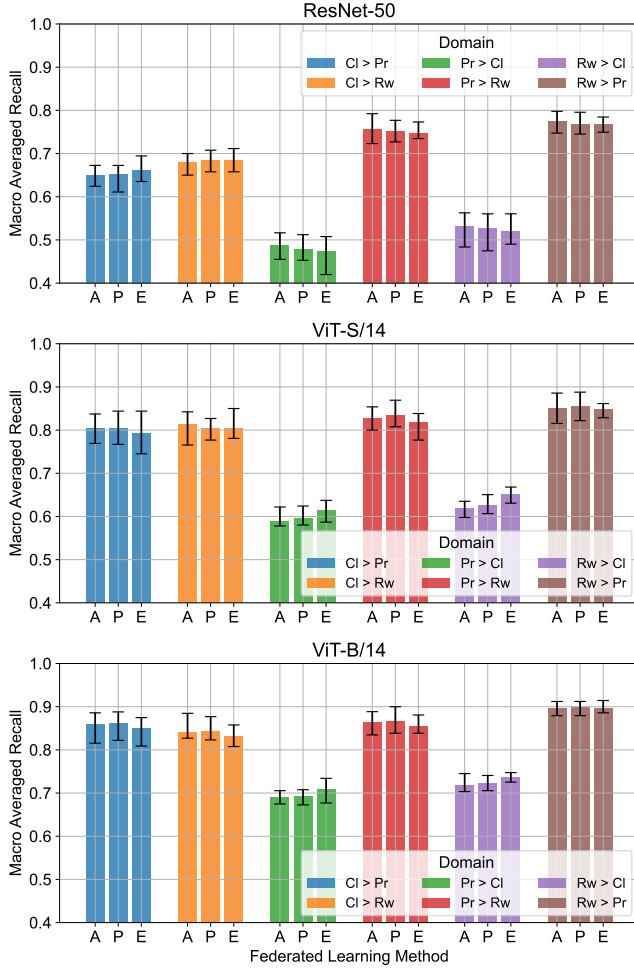
**Table 7:** FLOPs and model sizes for each model and training strategy.

Method	Model	FLOPs	Model Size
Fine-tuning	ResNet-50	24.6 G	94 MB
	ResNet-101	46.9 G	169 MB
Frozen VFMs	ViT-S	11.0 G	< 1 MB
	ViT-B	43.9 G	< 1 MB
Frozen VFMs with backbone skipped	ViT-S	0.61 M	< 1 MB
	ViT-B	1.20 M	< 1 MB

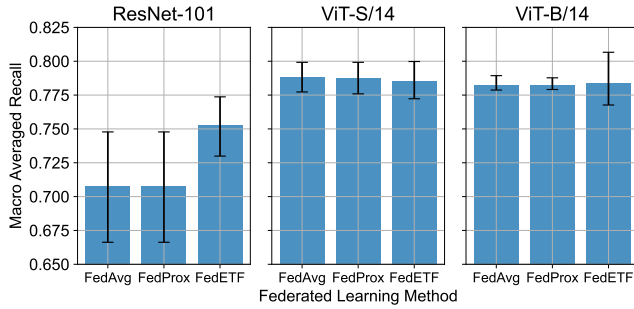
In the case of ResNet fine-tuning, training requires approximately three times the computational cost of a single forward pass. The resulting trained model, which is roughly 100 MB in size, must be transmitted. In contrast, when using frozen VFMs, the computational cost for ViT-S is reduced by half compared to ResNet-50. Moreover, only the bottleneck and classifier components, which together total less than 1 MB, need to be transmitted to the server, significantly improving communication efficiency. Furthermore, if the outputs of the frozen VFMs is stored in a feature bank during the initial training phase and the backbone computation is skipped thereafter, enabling training only of the bottleneck and classifier (Frozen VFMs with backbone skipped), it is possible to entirely eliminate backbone computation during training.

## E. REFERENCES

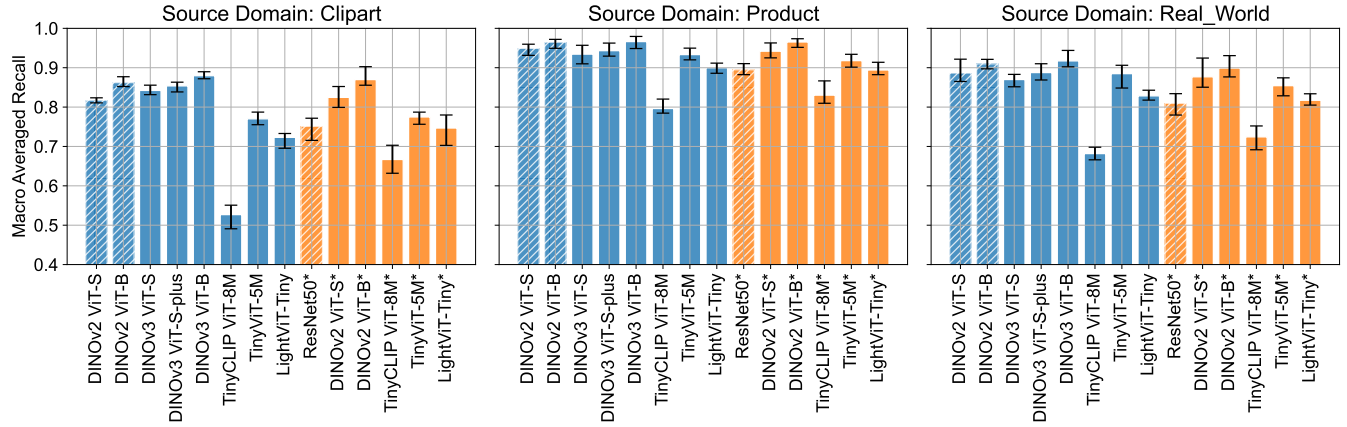
- [1] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “Randaugment: Practical automated data augmentation with a reduced search space,” in *CVPRW*, 2020, pp. 3008–3017.
- [2] T. Huang, L. Huang, S. You, F. Wang, C. Qian, and C. Xu, “Lightvit: Towards light-weight convolution-free vision transformers,” *arXiv preprint arXiv:2207.05557*, 2022.
- [3] K. Wu, J. Zhang, H. Peng, M. Liu, B. Xiao, J. Fu, and L. Yuan, “Tinyvit: Fast pretraining distillation for small vision transformers,” in *ECCV*, 2022.
- [4] K. Wu, H. Peng, Z. Zhou, B. Xiao, M. Liu, L. Yuan, H. Xuan, M. Valenzuela, X. S. Chen, X. Wang, H. Chao, and H. Hu, “Tinyclip: Clip distillation via affinity mimicking and weight inheritance,” in *Proc. of ICCV*, October 2023, pp. 21970–21980.
- [5] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, F. Massa, D. Haziza, L. Wehrstedt, J. Wang, T. Darcet, T. Moutakanni, L. Sentana, C. Roberts, A. Vedaldi, J. Tolan, J. Brandt, C. Couprie, J. Mairal, H. Jégou, P. Labatut, and P. Bojanowski, “DINOv3,” *arXiv preprint arXiv:2508.10104*, 2025.



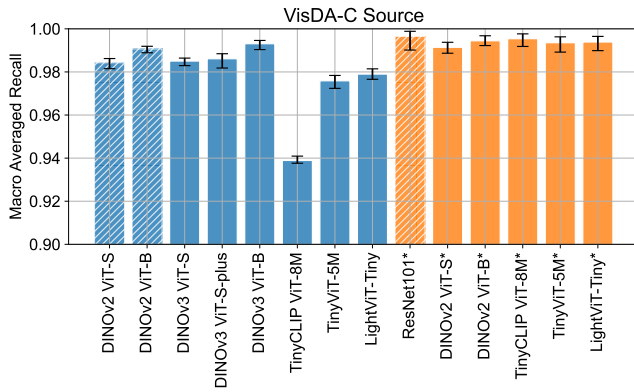
**Fig. 6:** The comparison among three different federated learning method, FedAvg (A), FedProx (P), and FedETF (E), with OH. Note that in this figure, colors indicate different source-target pairs. Each plot shows the average over nine runs, comprised of three different source imbalance ratios and three different target imbalance ratio. Error bars represent the maximum and minimum values from these nine runs.



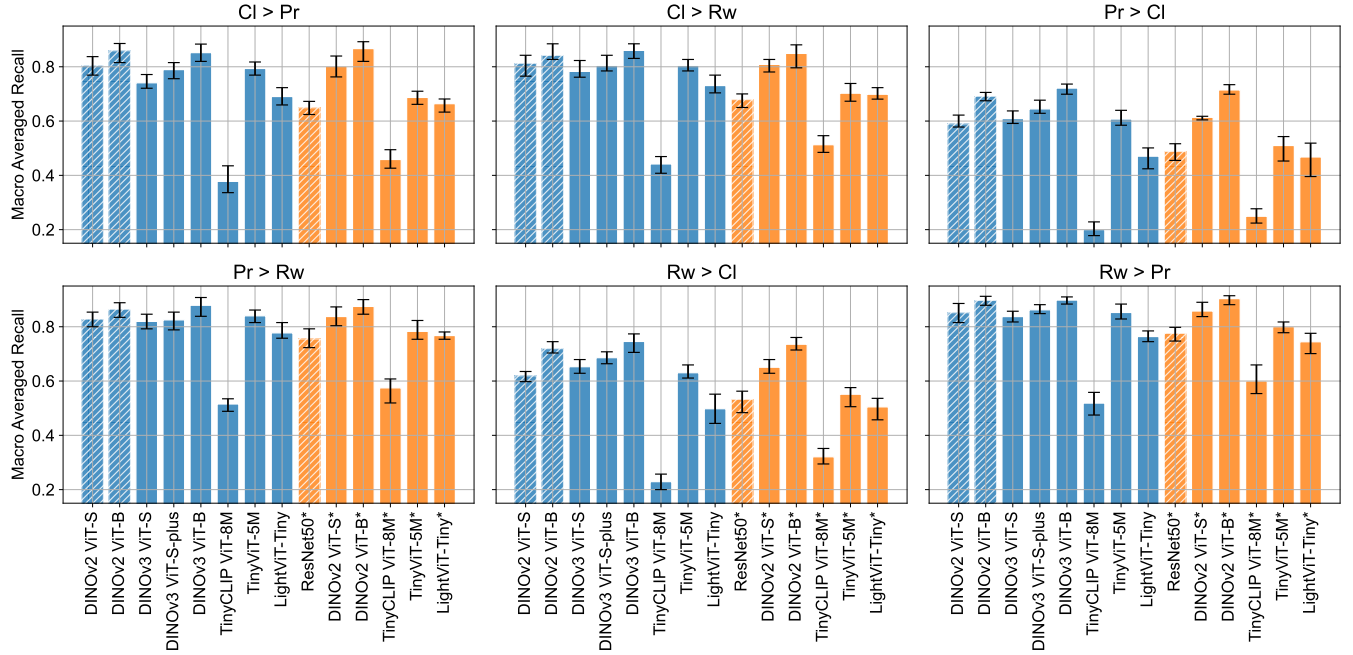
**Fig. 7:** The comparison among three different federated learning method, FedAvg, FedProx, and FedETF, with VisDA. Details of the plot are identical to those in Fig. 6.



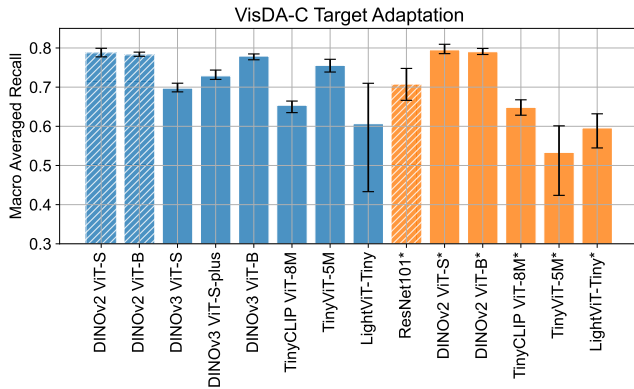
**Fig. 8:** Accuracy of the source domain on the OH dataset with various models. Blue bars indicate results obtained using a frozen backbone, while orange bars, denoted by an asterisk (\*) following the model name, represent results with a fine-tuned backbone. Hatched bars correspond to the original models discussed in this paper. Each plot is the average over nine runs, comprised of three different source imbalance ratios and three execution seeds for each source sampling. Error bars represent the maximum and minimum values of nine runs.



**Fig. 9:** Accuracy of the source domain on the VisDA dataset with various models. Details are same as Fig. 8.



**Fig. 10:** The whole results of federated target adaptation on the OH dataset with various models. Details of the bars are same as Fig. 8. Each plot shows the average over nine runs, comprised of three different source imbalance ratios and three different target imbalance ratio. Error bars represent the maximum and minimum values from these nine runs.



**Fig. 11:** Results of federated target adaptation on the VisDA dataset with various models. Details are same as Fig. 10.