Unified Alignment Protocol:
Making Sense of the Unlabeled Data in New Domains

Supplementary Material

8. Datasets

We assessed the performance of our proposed UAP on five widely used visual benchmarks commonly used for evaluating domain generalization methods. The details of these benchmark datasets are listed below.

PACS [40]: This dataset is a collection of 9,991 images with four distinct domains: art painting, cartoon, photo, and sketch. The task objective is classification across seven classes.

VLCS [8]: This dataset comprises 10,729 images spread across four domains, with each domain representing a distinct subdataset. The subdatasets include VOC2007, LabelMe, Caltech-101, and SUN09. The task objective is classification across five different classes.

OfficeHome [36]: OfficeHome dataset is a challenging benchmark composed of four visually distinct domains: Artistic images, Clipart images, Product images, and Real-world images. It comprises 15,500 images distributed across 65 object categories. The task objective is classification across these sixty five classes.

RotatedMNIST [32]: This dataset comprises MNIST images [5] that have been subjected to counter-clockwise rotations at angles of 0, 15, 30, 45, 60, and 75 degrees. These rotations result in six distinct domains: $M_0, M_{15}, M_{30}, M_{45}, M_{60}$, and M_{75}. The primary objective remains the classification of ten classes, corresponding to digits 0 through 9. We adopt the dataset variant used in [28, 30], where 1,000 images are rotated to define a domain.

TerraIncognita [1]: TerraIncognita dataset is a challenging benchmark composed of four visually distinct domains: L100, L38, L43 and L46. It comprises 24,788 images distributed across 10 classes. The task objective is classification across these ten classes.

9. Implementation Details

For performance evaluation, we allocated one domain as the server dataset and another as the unseen test domain for the final global model, assigning the remaining domains to individual clients. More concretely, in a dataset with M domains, one domain is used for the server, another for testing, and the rest, $M-2$ domains, are distributed among $M-2$ clients. This approach is similar to existing FDG methods [30], where each client possesses data from a unique domain. The accuracy of the final global model is then reported on the unseen test domain. For training, we set the batch size and initial learning rate at 64 and 0.002, respectively. We also set the number of local epochs to 5 and the total communication rounds to 40. After each communication round, client models are averaged using [25] method skipping the batch normalization parameters as done in [18]. For optimization, We utilized Stochastic Gradient Descent (SGD) as the optimizer and applied a cosine learning rate decay as the scheduler. The hyperparameters α and β are both set to 1, with λ set to 0.01 for all experiments. Ablation study of hyper-parameters (e.g., α, β, and λ) are reported in Ablation section.

Method	Test Domain	M_0					M_{15}				
	Server Domain	M_{15}	M_{30}	M_{45}	M_{60}	M_{75}	M_0	M_{30}	M_{45}	M_{60}	M_{75}
SSFL		77.40	56.00	37.00	23.50	15.10	68.50	67.20	41.30	38.70	36.80
UAP (Ours)		**81.90**	**65.10**	**55.70**	**34.90**	**20.90**	**84.30**	**87.00**	**63.40**	**52.90**	**31.00**

Table 9. *Performance comparison of baseline SSFL and UAP across two test domains (M_0, M_{15}) in the RotatedMNIST dataset. RotatedMNIST dataset consists of six domains: $M_0, M_{15}, M_{30}, M_{45}, M_{60}$ and M_{75}. For each combination, we allocated one domain as the server training dataset, another as the unseen test domain for the final global model while assigning the remaining domains to individual clients.*

Method	Test Domain	M_{30}					M_{45}				
	Server Domain	M_0	M_{15}	M_{30}	M_{60}	M_{75}	M_0	M_{15}	M_{30}	M_{60}	M_{75}
SSFL		43.00	71.60	75.10	54.80	41.30	32.00	50.20	76.20	70.60	53.50
UAP (Ours)		**59.40**	**87.90**	**84.10**	**64.20**	**47.90**	**47.40**	**64.70**	**89.80**	**88.40**	**66.30**

Table 10. *Performance comparison of baseline SSFL and UAP across two test domains (M_{30}, M_{45}) in the RotatedMNIST dataset. RotatedMNIST dataset consists of six domains: $M_0, M_{15}, M_{30}, M_{45}, M_{60}$ and M_{75}. For each combination, we allocated one domain as the server training dataset, another as the unseen test domain for the final global model while assigning the remaining domains to individual clients.*

10. Results on RotatedMNIST

The evaluation of UAP is presented in Tables 9, 10 and 11 on the RotatedMNIST dataset. There are 6 domains in RotatedMNIST dataset: $M_0, M_{15}, M_{30}, M_{45}, M_{60}$, and M_{75}. For reporting result of each combination, we allocated one domain as the server training dataset, another as the unseen test domain for the final global model while assigning the remaining domains to individual clients. In RotatedMNIST

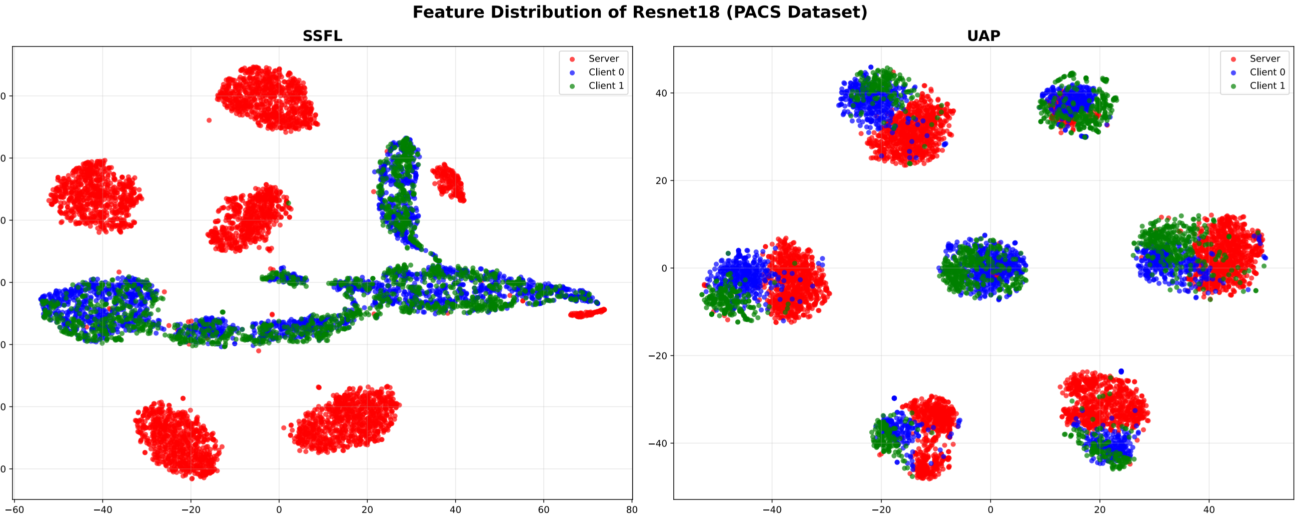

Figure 3. *Qualitative comparison of feature distribution of Resnet18 (T-SNE) of server and clients on PACS dataset [40].*

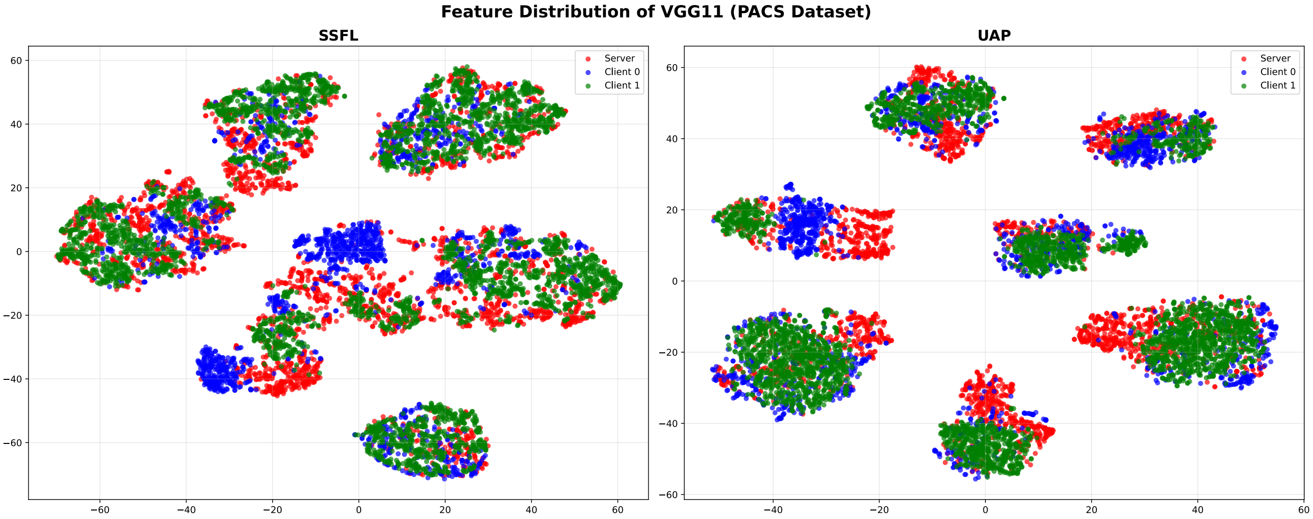

Figure 4. *Qualitative comparison of feature distribution of VGG11 (T-SNE) of server and clients on PACS dataset [40].*

dataset, we observe a consistent performance improvement with our proposed UAP over the baseline SSFL.

Method	Test Domain	M_{60}					M_{75}				
	Server Domain	M_0	M_{15}	M_{30}	M_{45}	M_{75}	M_0	M_{15}	M_{30}	M_{45}	M_{60}
SSFL		23.20	40.10	54.90	76.90	78.90	21.90	28.20	37.30	46.60	77.30
UAP (Ours)		**30.00**	**54.40**	**73.20**	**89.40**	**87.90**	**24.60**	**35.90**	**60.20**	**65.50**	**81.60**

Table 11. *Performance comparison of baseline SSFL and UAP across two test domains (M_{60}, M_{75}) in the RotatedMNIST dataset. RotatedMNIST dataset consists of six domains: $M_0, M_{15}, M_{30}, M_{45}, M_{60}$ and M_{75}. For each combination, we allocated one domain as the server training dataset, another as the unseen test domain for the final global model while assigning the remaining domains to individual clients.*

11. Comparison with SSFL and FDG Methods

Here, we compare our proposed UAP with SOTA SSFL methods [6, 16, 20] as well as SOTA FDG methods [22, 30, 34, 42]. We report performance of global model on the unseen Art domain of OfficeHome [40] dataset in Table 12 and on L100 test domain of TerraIncognita [1] dataset in Table 13. To train using FDG methods, we pretrain the server model with the server dataset and generate pseudo labels prior to client training. The results indicate that the current SSFL methods [6, 16, 20] struggles with Domain Generalization (DG). On the other hand, the FDG approaches are incapable of enhancing DG performance, even when trained using pseudo labels since these methods rely heavily on client-labeled data. Nonetheless, whereas existing SSFL and FDG methods struggles, our method thrives on them

Method	Clipart	GAIN	Product	GAIN
CBAFed [16]	35.68		30.94	
FedDG [22]	28.97		26.82	
FedDG-GA [42]	6.39		3.87	
FedGMA [34]	17.72		16.98	
FedSR [30]	4.80		5.27	
RScFed [20]	40.70		37.37	
SemiFL [6]	39.14		36.55	
UAP (Ours)	**48.95**	+8.25	**47.51**	+10.14

Table 12. *Comparative DG performance of SOTA SSFL and FDG methods and our proposed UAP on OfficeHome dataset. The table displays the results across two server training domains: Clipart and Product, with test performance of the global model reported on the unseen Art domain. The GAIN column shows performance improvement of our method compared to the second best method (highlighted by underline).*

Method	L38	GAIN	L43	GAIN
CBAFed [16]	35.45		46.25	
FedDG [22]	29.62		1.60	
FedDG-GA [42]	6.17		27.65	
FedGMA [34]	11.07		8.20	
FedSR [30]	8.99		46.22	
RScFed [20]	31.56		1.90	
SemiFL [6]	36.79		40.58	
UAP (Ours)	**40.17**	+3.38	**48.64**	+2.39

Table 13. *Comparative DG performance of SOTA SSFL and FDG methods and our proposed UAP on TerraIncognita dataset. The table displays the results across two server training domains: L38 and L43, with test performance of the global model reported on unseen L100 domain. The GAIN column shows performance improvement of our method compared to the second best method (highlighted by underline).*

and successfully generalizes across domains. These results successfully establish the significance of our proposed UAP for achieving S-FDG.

12. Abltation Study

All our ablation studies for hyperparameters are conducted using the PACS [40] benchmark dataset, with Art Painting as the unseen test domain and Cartoon and Photo as the server domains.

Effect of different α **&** β: In Table 14a, we present the impact of varying α and β respectively. From the results, we find that a value of 1 for these parameters delivers optimal results, with any deviation leading to suboptimal performance. The empirical data presented in this table justifies our selection of the hyperparameters α and β.

Effect of λ: We report the effect of changing hyperparameter λ in Table 14b. The results confirm that a value of $\lambda = 0.01$ results in optimal performance. Thus justifying our choice of hyperparameter λ.

Effect of Reference matrix: We report the effect of changing reference matrix Σ in Table 15. We experiment by setting value of $\Sigma = \gamma \mathbf{\Sigma}_k$ by varying γ to 50, 100 and 200. We also experiment with minimizing the offdiagonal elements of covariance matrices to 0 without constraining the diagonal elements. The results confirm that diagonal matrix with a value of $\gamma = 100$ results in optimal performance. Thus justifying our choice of hyperparameter σ.

α, β	Cartoon	Photo
0.5, 0.5	75.49	61.52
1.0, 1.0	**75.73**	**64.40**
2.0, 2.0	75.34	62.84

(a) Effect of α and β

λ	Cartoon	Photo
0.0001	68.55	57.37
0.01	**75.73**	**64.40**
1.0	72.51	57.57

(b) Effect of λ

Table 14. *Ablation studies on the effect of jointly changing* α *and* β *and varying* λ *in the PACS dataset. We report the performance on two server domains, Cartoon and Photo and testing on Art Painting domain, with different values of these hyperparameters.*

$\mathcal{L}_{COV}$	γ	Cartoon	Photo	Sketch
	0.5	**78.32**	58.98	66.94
$\frac{1}{m}\lVert\mathbf{\Sigma}_z - \gamma\mathbf{\Sigma}_k\rVert^2$	1.0	75.73	**64.40**	**67.92**
	2.0	68.36	51.47	63.92
$\frac{1}{m}\sum_{i\neq j}[\mathbf{\Sigma}_z]_{ij}^2$	-	74.17	63.04	59.96

Table 15. *Ablation study on the effect of changing the reference matrix in the PACS dataset. We report the performance on three server domains (Cartoon, Photo, and Sketch) when testing on the Art Painting domain.*

13. Remaining Results

PACS: The evaluation of UAP is presented in Table 16 on the PACS dataset. From the results we see that the generalization performance of UAP degrades slightly with sketch as test domain. Again this can be attributed to the weaker feature alignment of the remaining domains with sketch domain. Nevertheless, similar to other datasets, we observe a consistent improvement in performance on the Photo test domain with proposed UAP compared to baseline SSFL.

VLCS: The evaluation of UAP is presented in Table 17 on the VLCS dataset. Similarly to other data sets, we observe performance improvement with our proposed UAP over the baseline SSFL.

OfficeHome: The evaluation of UAP is presented in Table 18 on the OfficeHome dataset. Similarly to other data sets, we observe performance improvement with our proposed UAP over the baseline SSFL.

Method	Unseen Test Domain	**P**			**S**		
	Server Trained on	A	C	S	A	C	P
SSFL		86.05	82.93	32.34	**65.89**	**75.36**	30.67
UAP (Ours)		**87.31**	**86.41**	**79.76**	64.06	71.57	**32.50**

Table 16. *Performance comparison of baseline SSFL and UAP across different test domains P and S in the PACS dataset.*

Method	Unseen Test Domain	**L**			**S**		
	Server Trained on	C	S	V	C	L	V
SSFL		46.99	54.37	56.40	44.45	**65.39**	67.28
UAP (Ours)		**48.49**	**58.58**	**58.73**	**50.24**	52.86	**70.87**

Table 17. *Performance comparison of baseline SSFL and UAP across various test domains L and S within the VLCS dataset [8].*

Method	Unseen Test Domain	**P**			**R**		
	Server Trained on	A	C	R	A	C	P
SSFL		52.85	54.88	70.74	61.28	55.80	60.98
UAP (Ours)		**55.96**	**54.92**	**72.63**	**63.92**	**59.49**	**64.65**

Table 18. *Performance comparison of baseline SSFL and UAP across different test domains P and R in the OfficeHome dataset [36].*