A. Hyperparameter setting

We report the detailed hyperparameters setting with a specific model for each dataset in Table 8 and Table 9.

A.1. Setup for Table 2

For classic CV tasks, we follow the setup from the original papers using USB codebase. The details setup hyperparameters are listed in Table 8.

A.2. Setup for Table 7

Pre-trained ViT models [18] are used for CV tasks in USB. For TissueMNIST, CIFAR-100, and Euro-SAT, we use ViT-Tiny and ViT-Small with a patch size of 4 and an image size of 32, while for Semi-Aves, we use ViT-Small with a patch size of 16 and an image size of 224. For STL10, which is a subset of ImageNet, we use unsupervised pre-training MAE [21] of ViT-Base with an image size of 96 to prevent cheating.

Following USB CV tasks, we adopt layer-wise learning rate decay as in [31]. The cosine annealing scheduler is used with a total step of 204,800 and warm-up for 5,120 steps. Both labeled and unlabeled batch sizes are set to 16, and other algorithm-related hyper-parameters remain the same as in the original papers.

B. ImageNet detailed results

Table 10 shows the detailed results from Table 3. EPASS achieves 75.3% of top-1 accuracy with the same training duration ($\sim$ 400 epochs) on 10% of labels for SimMatch, and 74.1% of top-1 accuracy for CoMatch. These improvements are also noticeable when EPASS is deployed on 1% of labels, achieving 67.4% and 68.6% top-1 accuracy for CoMatch and SimMatch, respectively.

C. Precision, Recall, F1 and AUC

We further report precision, recall, F1-score, and AUC (area under curve) results on the CIFAR-10/100, SVHN, and STL-10 datasets. As shown in Table 11 and Table 12, EPASS also has the best performance on precision, recall, F1-score, and AUC on all datasets except CIFAR. Especially on the STL-10 dataset, the improvement from EPASS for CoMatch and SimMatch is very noticeable by a large margin.

D. List of Data Transformations

We report the detailed augmentations used in our method in Table 13. This list of transformations is similar to the original list used in FixMatch [40] and FlexMatch [52].

Dataset	CIFAR-10	CIFAR-100	STL-10	SVHN	ImageNet
Model	WRN-28-2	WRN-28-8	WRN-37-2	WRN-28-2	ResNet-50
Weight Decay	5e-4	1e-3	5e-4	5e-4	3e-4
Labeled Batch Size	64				128
Unlabeled Batch Size	448				128
Learning Rate	0.03				
SGD Momentum	0.9				
EMA Momentum	0.999				
Scheduler	$\eta = \eta_0 \cos\left(\frac{7\pi k}{16K}\right)$				
Weak Augmentation	Random Crop, Random Horizontal Flip				
Strong Augmentation	RandAugment [16]				
Unsupervised Loss Weight	1				

Table 8. Dataset-wise hyperparameters for classic CV tasks.

Dataset	CIFAR-100	STL-10	Euro-SAT	TissueMNIST	Semi-Aves
Image Size	32	96	32	32	224
Model	ViT-S-P4-32	ViT-B-P16-96	ViT-S-P4-32	ViT-T-P4-32	ViT-S-P16-224
Weight Decay	5e-4				
Labeled Batch Size	16				
Unlabeled Batch Size	16				
Learning Rate	5e-4	1e-4	5e-5	5e-5	1e-3
Layer Decay Rate	0.5	0.95	1.0	0.95	0.65
Scheduler	$\eta = \eta_0 \cos\left(\frac{7\pi k}{16K}\right)$				
Model EMA Momentum	0.0				
Prediction EMA Momentum	0.999				
Weak Augmentation	Random Crop, Random Horizontal Flip				
Strong Augmentation	RandAugment [16]				

Table 9. Dataset-wise hyperparameters for USB [42] CV tasks.

Self-supervised Pre-training	Method	Epochs	Parameters (train/test)	1% labels top-1	1% labels top-5	10% labels top-1	10% labels top-5
None	FixMatch	$\sim$ 300	25.6M/25.6M	-	-	71.5	89.1
	CoMatch [30]	$\sim$ 400	30.0M/25.6M	66.0	86.4	73.6	91.6
	SimMatch [56]	$\sim$ 400	30.0M/25.6M	67.2	87.1	<u>74.4</u>	<u>91.6</u>
MoCo V2 [13]	CoMatch [30]	$\sim$ 1200	30.0M/25.6M	67.1	87.1	73.7	91.4
MoCo-EMAN [8]	FixMatch-EMAN [8]	$\sim$ 1100	30.0M/25.6M	63.0	83.4	74.0	90.9
None	**[30] + EPASS**	$\sim$ 400	30.0M/25.6M	<u>67.4</u>	<u>87.3</u>	74.1	91.5
None	**[56] + EPASS**	$\sim$ 400	30.0M/25.6M	**68.6**	**87.6**	**75.3**	**92.6**

Table 10. Accuracy results on ImageNet with **1%** and **10%** labeled examples.

Dataset	CIFAR-10 (40)			CIFAR-100 (400)		
Criteria	Precision	Recall	F1 Score	Precision	Recall	F1 Score
UDA	0.9333	0.9311	0.9302	0.5813	0.5484	0.5087
FixMatch	0.9351	0.9307	0.9297	0.5574	0.5430	0.4946
Dash	0.8847	0.8486	0.8210	0.5833	0.5649	0.5215
FlexMatch	0.9505	0.9507	0.9505	0.6135	0.6193	0.6107
FreeMatch	**0.9510**	**0.9512**	**0.9510**	**0.6243**	**0.6261**	**0.6137**
CoMatch	0.9441	0.9445	0.9441	0.4543	0.3979	0.4067
SimMatch	0.9434	0.9438	0.9434	0.5101	0.5133	0.5017
[30] + EPASS	0.9447	0.9450	0.9447	0.5588	0.4927	0.4978
[56] + EPASS	0.9493	0.9494	0.9491	0.6084	0.6061	0.6003

Table 11. Precision, recall, F1-score and AUC results on CIFAR-10/100.

Dataset	SVHN (40)			STL-10 (40)		
Criteria	Precision	Recall	F1 Score	Precision	Recall	F1 Score
UDA	0.9781	0.9777	0.9780	0.6385	0.5319	0.4765
FixMatch	0.9731	0.9706	0.9716	0.6590	0.5830	0.5405
Dash	0.9779	0.9777	0.9778	0.8117	0.6020	0.5448
FlexMatch	0.9566	0.9691	0.9625	0.6403	0.6755	0.6518
FreeMatch	0.9551	0.9665	0.9605	0.8489	0.8439	0.8354
CoMatch	0.9542	0.9677	0.9605	-	-	-
SimMatch	0.9718	0.9782	0.9748	-	-	-
[30] + EPASS	0.9647	0.9724	0.9684	**0.9100**	**0.9085**	**0.9075**
[56] + EPASS	**0.9782**	**0.9778**	**0.9780**	0.8026	0.8029	0.7977

Table 12. Precision, recall, F1-score and AUC results on SVHN and STL-10.

Transformation	Description	Parameter	Range
Autocontrast	Maximizes the image contrast by setting the darkest (lightest) pixel to black (white).		
Brightness	Adjusts the brightness of the image. $B = 0$ returns a black image, $B = 1$ returns the original image.	B	[0.05, 0.95]
Color	Adjusts the color balance of the image like in a TV. $C = 0$ returns a black & white image, $C = 1$ returns the original image.	C	[0.05, 0.95]
Contrast	Controls the contrast of the image. A $C = 0$ returns a gray image, $C = 1$ returns the original image.	C	[0.05, 0.95]
Equalize	Equalizes the image histogram.		
Identity	Returns the original image.		
Posterize	Reduces each pixel to B bits.	B	[4, 8]
Rotate	Rotates the image by θ degrees.	θ	[-30, 30]
Sharpness	Adjusts the sharpness of the image, where $S = 0$ returns a blurred image, and $S = 1$ returns the original image.	S	[0.05, 0.95]
Shear_x	Shears the image along the horizontal axis with rate R.	R	[-0.3, 0.3]
Shear_y	Shears the image along the vertical axis with rate R.	R	[-0.3, 0.3]
Solarize	Inverts all pixels above a threshold value of T.	T	[0, 1]
Translate_x	Translates the image horizontally by ($\lambda \times$image width) pixels.	λ	[-0.3, 0.3]
Translate_y	Translates the image vertically by ($\lambda \times$image height) pixels.	λ	[-0.3, 0.3]

Table 13. List of transformations used in RandAugment

E. Qualitative Analysis

We present the T-SNE visualization of features on STL-10 test dataset with 40-label split in Figure 5,6. The visualization is using trained models from SimMatch and CoMatch with EPASS.

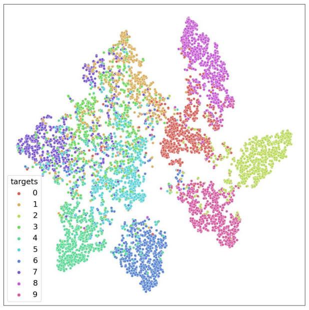

Figure 5. T-SNE visualization of SimMatch + EPASS features on STL-10 dataset with 40-label split.

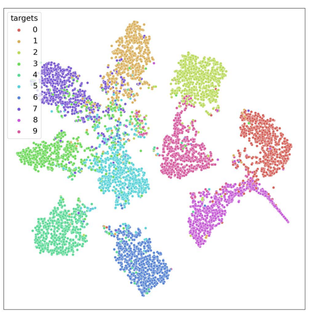

Figure 6. T-SNE visualization of CoMatch + EPASS features on STL-10 dataset with 40-label split.

We also illustrate the T-SNE visualization of features on SVHN test dataset and CIFAR-10 test dataset with 40-label split in Figure 7,8 and Figure 9,10, respectively.

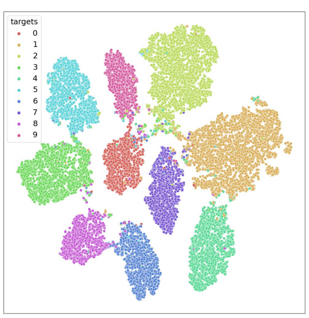

Figure 7. T-SNE visualization of SimMatch + EPASS features on SVHN dataset with 40-label split.

Furthermore, we sketch the T-SNE visualization for the embeddings on those three datasets, as shown in Figures 11,

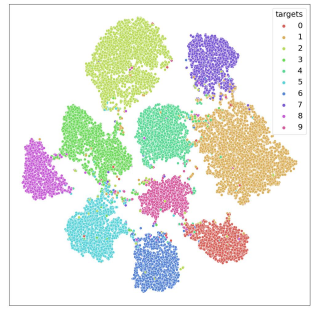

Figure 8. T-SNE visualization of CoMatch + EPASS features on SVHN dataset with 40-label split.

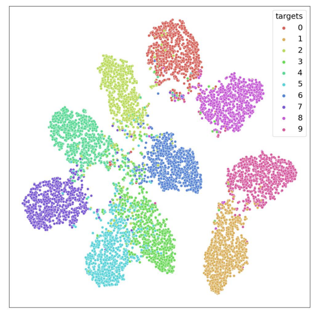

Figure 9. T-SNE visualization of SimMatch + EPASS features on CIFAR-10 dataset with 40-label split.

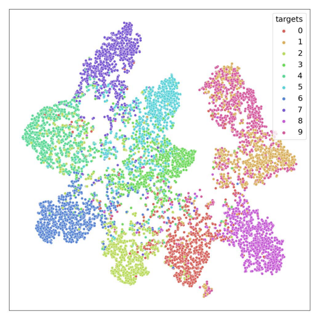

Figure 10. T-SNE visualization of CoMatch + EPASS features on CIFAR-10 dataset with 40-label split.

12, 13, 14, 15, 16, respectively.

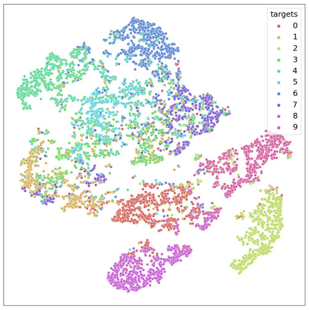

Figure 11. T-SNE visualization of SimMatch + EPASS embeddings on STL-10 dataset with 40-label split.

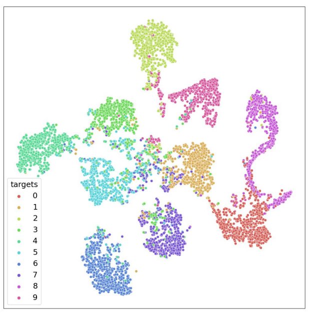

Figure 12. T-SNE visualization of CoMatch + EPASS embeddings on STL-10 dataset with 40-label split.

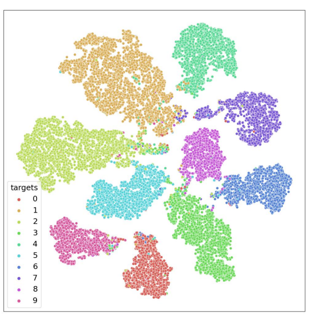

Figure 13. T-SNE visualization of SimMatch + EPASS embeddings on SVHN dataset with 40-label split.

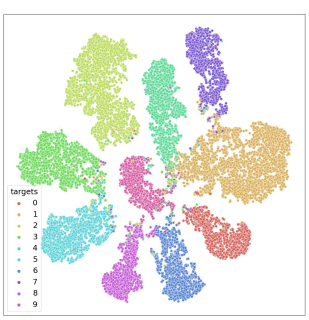

Figure 14. T-SNE visualization of CoMatch + EPASS embeddings on SVHN dataset with 40-label split.

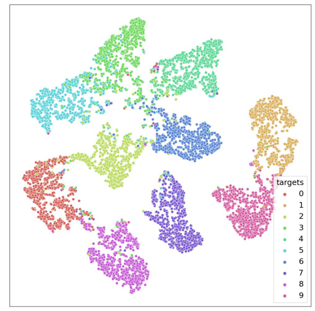

Figure 15. T-SNE visualization of SimMatch + EPASS embeddings on CIFAR-10 dataset with 40-label split.

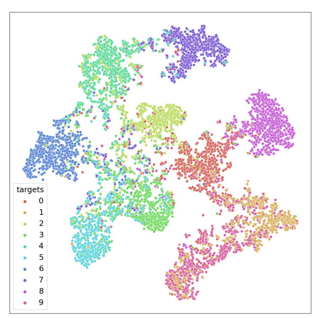

Figure 16. T-SNE visualization of CoMatch + EPASS embeddings on CIFAR-10 dataset with 40-label split.

F. Algorithm

We apply EPASS to recent state-of-the-art SSL (CoMatch [30] and SimMatch [56]) and self-supervised learning (MoCo [22]). Applying EPASS to these methods only requires a few lines of code as shown in Algorithm 1.

Input: Encoder f, projector g_k and the number of projectors K.

for $b = 1$ **to** μB **do**

  Generate prediction distribution as a conventional pipeline by forward propagation.

  for $k = 1$ **to** K **do**

    $z_{b,k} = g_k\left(f_b\right)$ `// Compute embeddings by different projectors.`

  end

  $z_b = norm\left(\frac{\sum_{k=1}^{K} z_{b,k}}{K}\right)$ `// Compute the aggregated embeddings.`

  Calculate the overall training objective.

  Optimize the model and update the memory bank.

end

Output: The optimized model f_s, h_s and $g_{s,k}$.

Algorithm 1: EPASS