Appendix

1. Proof of Eq. 2

From the definition of KL divergence, Eq. 2 can be formulated as follows:

$$
\begin{aligned}
&KL(P_p(x)||\hat{P}_p(x)) \\
&= E_p[\log(\frac{P_p(x)}{\hat{P}_p(x)})] \\
&= E_p[\log(f^*(x))] - \log(E_u[f^*(x)]) \\
&\quad - E_p[\log(f(x))] + \log(E_u[f(x)]) \\
&= \mathcal{L}_{var}(f(x)) - \mathcal{L}_{var}(f^*(x)),
\end{aligned}
$$

where

$$\mathcal{L}_{var}(f(x)) = \log(E_u[f(x)]) - E_p[\log(f(x))].$$

2. Proof of Eq. 16

Given that $0<\sum_{i=1}^{n_u} f(x_i^u)<n_u$, then let

$$S = \frac{1}{n_u}\sum_{i=1}^{o}\sigma_u^{i-1}, \tag{21}$$

and

$$\sigma_u S = \frac{1}{n_u}\sum_{i=1}^{o}\sigma_u^{i}. \tag{22}$$

Let Eq. 21-Eq. 22, then

$$(1-\sigma_u)S = \frac{1}{n_u} - \frac{1}{n_u}\sigma_u^o,$$

and then,

$$S = \frac{1-\sigma_u^o}{n_u(1-\sigma_u)},$$

and

$$\frac{\partial \hat{\mathcal{L}}_{Tar-u} f(x)}{\partial \theta} = \frac{1-\sigma_u^o}{\sum_{i=1}^{n_u} f(x_i^u)}\sum_{i=1}^{n_u}\nabla_\theta f(x_i^u).$$

3. Training Details for *T-HOneCls*

Training Details A detailed description of the *self-calibrated optimization* is provided in Algorithm 1. The hyperspectral image classification is a one-shot image input. Stochastic gradient descent degenerates into gradient descent in the process of network optimization. A global proportional random stratified sampler (the sampling operation in Algorithm 1) is also proposed to recover the stochastic gradient descent. The detailed sampling algorithm is described in the following:

Algorithm 1: Self-calibrated optimization

```
Input: $H$ : hyperspectral imagery; $M_{in}$ : a set of
       training masks; $o$ : the order of the Taylor
       series; $\alpha$ : smoothing factor; $n_{pb}$ : number of
       pseudo batches; $T$ : training epochs; $S_{net}$ :
       student network; $T_{net}$ : teacher network.
Output: The weight of the teacher network
Initialize the weight of the student network ($\theta_S$) and
 the teacher network ($\theta_T$)
for t=1 to T do
    $M_{out}$ =Sampling($M_{in}, n_{pb}$)
    for e=1 to $n_{pb}$ do
        $p_S = S_{net}(H)$
        $p_T = T_{net}(H)$
        $\mathcal{L}_S = \mathcal{L}_{Tar}(p_S, M_{out}[0][e], M_{out}[1][e])$
           $+\beta\mathcal{L}_{kl}(p_S, p_T, M_{out}[0][e], M_{out}[1][e])$
        update $\theta_S$
        update $\theta_T$ : $\theta_T^e = \alpha\theta_T^{e-1} + (1-\alpha)\theta_S^e$
```

Algorithm 2: Global proportional random stratified sampling

```
Input: $M_{in} = \{m_{in}^i\}_{i=0}^1$: a set of training masks;
       $n_{pb}$:Number of pseudo batches.
Output: $M_{out}$: a list of sets of stratified masks
$M_{out} \leftarrow []$ // Initialize an empty list
for k=0 to 1 do
    $I_k \leftarrow \{j | m_{in}^{kj} = 1\}$
    $I_k \leftarrow$ Random shuffle($I_k$)
    $M_{out}[k] \leftarrow []$
    $L_k = |I_k| // n_{pb}$
    while $|I_k| \geq L_k$ do
        $r \leftarrow I_k.pop(L_k)$
        // Fetch $L_k$ samples from $I_k$
        $M_{out}[k].push(r)$
```

Global Proportional Random Stratified Sampler Stochastic gradient descent is the mainstream optimization approach at present, so some objective functions are used based on stochastic gradient descent [21, 3, 10, 42, 35, 7]. Whether there will be a problem when these objective functions encounter gradient descent is not clear. As the *Taylor variational loss* can be optimized not only using stochastic gradient descent, but also using gradient descent based optimization methods, in order to ensure the adaptability of the proposed framework to different objective functions, we propose the global proportional random stratified sampler, which can recover the stochastic gradient descent by constructing pseudo-batches (such as

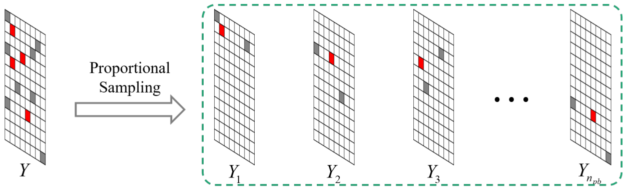


Figure 6: The description of Global proportional random stratified sampler.

Fig. 6).

The proposed sampler is summarized in Algorithm. 2. The input of this sampler is a positive mask (m_{in}^1) and an unlabeled mask (m_{in}^0), and the data used for training are labeled as 1 and the other data are labeled as 0. The key idea of the proposed sampler is to randomly train $|I_1|//n_{pb}$ positive samples and $|I_2|//n_{pb}$ unlabeled samples each time (stratified), where each batch has both positive samples and unlabeled samples (proportional). By constructing pseudo-batches, we can meet the requirements of the current objective function for stochastic gradients. The output of the sampler is a list of positive and unlabeled masks, with the data used for training in each batch labeled with 1 and the rest labeled with 0.

4. The Description of Datasets and Hyperparameters

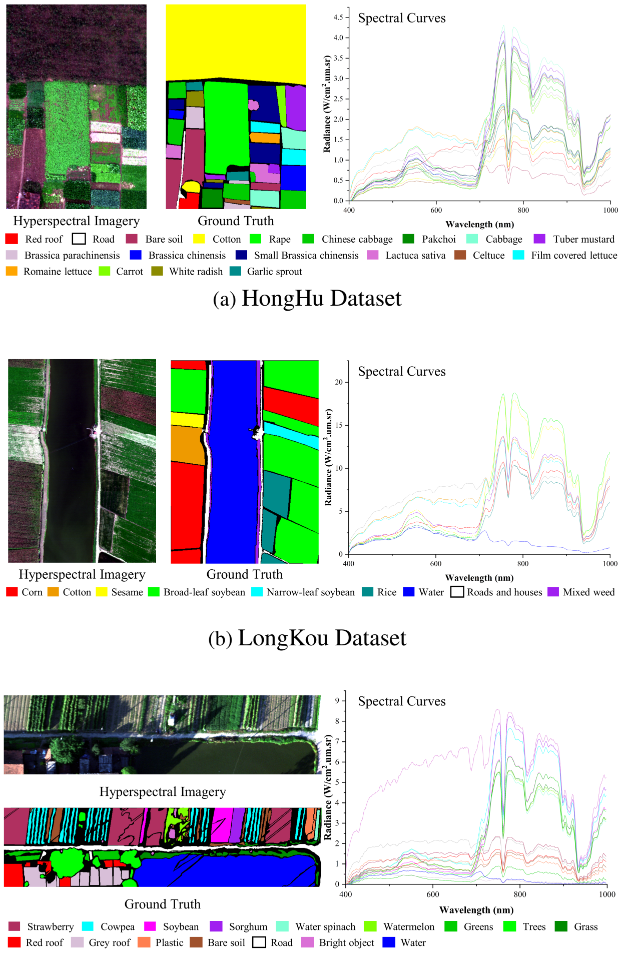
The HongHu, LongKou and HanChuan HSIs, along with the ground truth and spectral curves as examples, are shown in Fig. 7. It can be seen from the Fig. 7 that the spectral curves of vegetation are very similar, and it is very challenging to identify the specific vegetation types. The hyperparameters were shown in Table 7-Table 9.

5. The Structure of FreeOCNet

The FreeOCNet includes encoder, decoder and lateral connection (Fig. 8). The basic module in encoder is a spectral-spatial-attention (SSA)-convolution layer (Conv 3)-Group normalization-rectified linear unit (ReLU), and the mdoule of a Conv 3 × 3 with stride 2 to reduce the spatial size. A lightweight decoder is used, which consists of a Conv 3 × 3 layer and 2× upsampling layer and a fixed number of channels. A Conv 1 × 1 layer is used in lateral connection to reduce the number of channels in the encoder.

6. More Experimental Results

The distribution maps for the HongHu, LongKou and HanChuan datasets are shown in Fig. 9a, Fig. 9b and Fig. 9c, respectively. The Precision and Recall for the HongHu, HanChuan and LongKou datasets are shown in Table 10, Table 11 and Table 12, respectively. As shown in this subsection, other methods cannot obtain high precision and recall at the same time, that is, these methods cannot find a balance between the overfitting and underfitting of the training data. This balance was found by *T-HOneCls*, and a good F1-score was obtained by *T-HOneCls* in all tasks.

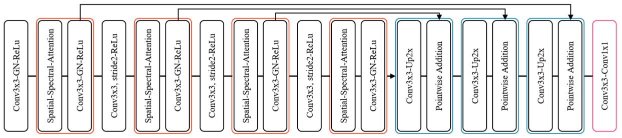


(a) HongHu Dataset

(b) LongKou Dataset

(c) HanChuan Dataset

Figure 7: UAV HSIs with ground truth and spectral curves.



Stem	Encoder	Lateral	Decoder
Conv 3x3:64 Groups:16	Conv 3x3 #1-#4:128,192,256,320 Groups:16 Conv 3x3, stride 2,#1-#3:192,256,320	Conv 1x1:128	Conv 3x3:128

Figure 8: The description of FreeOCNet.

7. More Experimental Results for the Training Process and Training Samples

The curves of the positive class and the total loss of the different positive training samples of rape and cabbage are

Dataset	Classes selected for classification	Labeled samples for each class	Unlabeled samples for each class	Validation samples for each class	Hyperparameters
HongHu (270 channels)	Cotton, Rape, Chinese cabbage, Cabbage, Tuber mustard	100	4000	290878	Pseudo batch number: 10 Epochs:150 Optimizer: SGD (lr=0.0001, momentum=0.9, weight_decay=0.0001) with ExponentialLR (gamma=0.995)
LongKou (270 channels)	Corn, Sesame, Broad-leaf soybean, Rice	100	4000	203642	Pseudo batch number: 10 Epochs: 150 Optimizer: SGD (lr=0.0001, momentum=0.9, weight_decay=0.0001) with ExponentialLr (gamma=0.995)
HanChuan (274 channels)	Strawberry, Cowpea, Soybean, Watermelon, Road, Water	100	4000	255930	Pseudo batch number: 10 Epochs: 170 Optimizer: SGD (lr=0.0002, momentum=0.9, weight_decay=0.0001) with ExponentialLr (gamma=0.995)

Table 7: Details of the UAV hyperspectral datasets and hyperparameters

Dataset	Classes	Labeled samples for each class	Unlabeled samples for each class	Validation samples for each class	Hyperparameters
India Pines (200 channels)	2,11	100	4000	10149	Pseudo batch number:10 Epoch:300 Optimizer:SGD(lr=0.0001,momentum=0.9,weight_decay=0.0001) with ExponentialLR(gamma=0.995)
Pavia University (103 channels)	2	100	4000	42676	Pseudo batch number:10 Epoch:100 Optimizer:SGD(lr=0.0001,momentum=0.9,weight_decay=0.0001) with ExponentialLr(gamma=0.995)
	8	100	4000	42676	Pseudo batch number:10 Epoch:300 Optimizer:SGD(lr=0.0001,momentum=0.9,weight_decay=0.0001) with ExponentialLr(gamma=0.995)

Table 8: Details of the India Pines and Pavia University datasets and hyperparameters

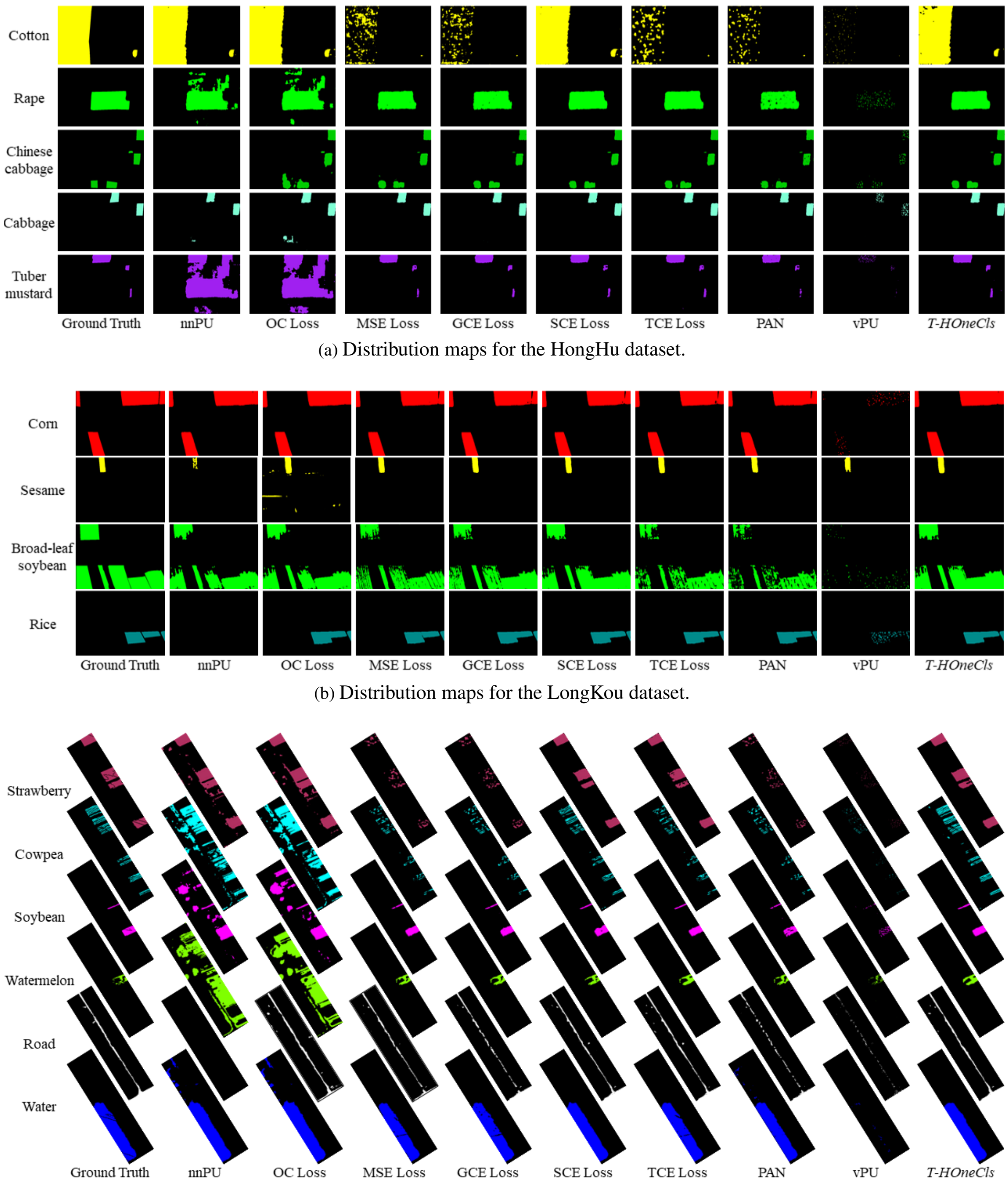
Dataset	Classes	Labeled samples for positive class	Unlabeled samples	Validation samples	Hyperparameters
CIFAR-10	Positive:0,1,8,9 Negative:2,3,4,5,6,7	900	45000	10000	Epoch:50 Order:1 Optimizer:Adam(lr=3e-5,betas=(0.5, 0.99))
STL-10	Positive:0,2,3,8,9 Negative:1,4,5,6,7	900	99000	8000	Epoch:50 Order:3 Optimizer:Adam(lr=3e-5,betas=(0.5,0.99))

Table 9: Details of the CIFAR-10 and STL-10 datasets and hyperparameters

Class	Class prior-based classifiers		Label noise representation learning				Class prior-free classifiers		
	nnPU	OC Loss	MSE Loss	GCE Loss	SCE Loss	TCE Loss	PAN	vPU	T-HOneCls
Cotton	99.34/99.54	99.31/**99.57**	99.98/9.52	**100.00**/10.19	99.94/93.08	**100.00**/11.29	**100.00**/9.09	99.94/0.94	99.96/96.40
Rape	69.79/99.58	69.38/**99.71**	99.86/93.03	99.86/93.73	99.88/94.95	99.78/95.59	**99.93**/64.81	99.74/4.34	99.77/95.92
Chinese cabbage	0.00/0.00	81.19/**96.29**	95.96/91.40	97.81/90.60	97.58/90.27	97.31/91.27	**98.19**/87.11	97.01/14.28	95.97/92.60
Cabbage	54.18/54.48	81.55/**99.91**	99.87/98.54	99.89/98.32	99.89/98.37	99.85/98.75	**99.92**/96.50	99.88/21.12	99.79/98.95
Tuber mustard	13.63/99.73	13.36/**99.88**	99.00/91.75	98.68/93.57	98.72/92.49	98.41/94.87	**99.33**/86.00	99.30/13.19	98.56/96.24

Table 10: The Precision/Recall for the HongHu dataset

shown in Fig. 10 and Fig. 11, respectively. The curves of the F1-score are also shown. The variational loss using fewer training samples leads to the gradient domination optimization process of unlabeled samples at the beginning of the training, which makes the loss of positive classes rise at the beginning of the training. Although the loss of the positive samples will decreases as the training progresses, for example 40, 100 or 400, the F1-score is unstable, and determining the optimal training epochs is very challenging without using additional data. In the classification of rape in the HongHu dataset, 4000 positive training samples can obtain a stable F1-score, but the F1-score of cabbage is still unstable. However, this shortcoming is overcome by the proposed *T-HOneCls*, and a stable F1-score can be obtained, as shown in Fig. 10f and Fig. 11f.



(a) Distribution maps for the HongHu dataset.

(b) Distribution maps for the LongKou dataset.

(c) Distribution maps for the HanChuan dataset.

Figure 9: Distribution maps for the UAV hyperspectral datasets. The maps with the best F1-score are displayed for five experiments.

8. More Experimental Results for the Order of the Taylor Series

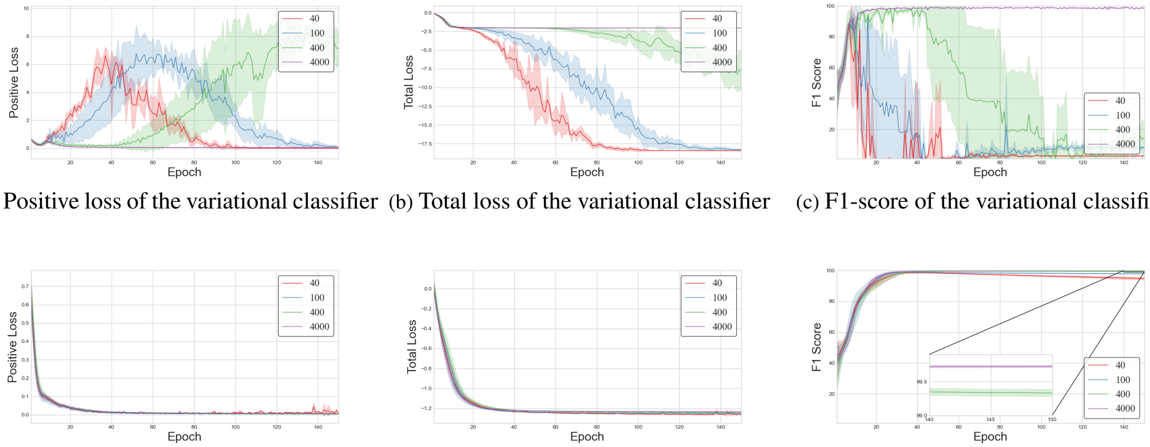
The results for cotton and five other ground objects are displayed in Fig. 12. The most important contribution of this paper is to point out that the reason for the poor performance of variational loss is that the gradient of the unlabeled data is given too much weight. To solve this problem, Taylor expansion is introduced in the variational loss, so as

Class	Class prior-based classifiers		Label noise representation learning				Class prior-free classifiers		
	nnPU	OC Loss	MSE Loss	GCE Loss	SCE Loss	TCE Loss	PAN	vPU	T-HOneCls
Strawberry	80.94/99.30	81.22/**99.75**	**99.85**/20.38	99.76/20.93	99.78/86.12	99.81/66.12	99.74/18.32	98.50/4.94	99.16/90.44
Cowpea	42.79/**98.91**	42.12/98.61	99.83/30.40	**99.94**/30.13	98.90/55.82	99.83/39.77	99.89/31.37	99.04/6.86	96.69/84.77
Soybean	27.96/99.85	26.86/**99.98**	**99.69**/95.27	99.51/95.13	99.43/95.07	99.56/97.55	98.93/77.48	96.86/24.22	99.62/98.64
Watermelon	6.25/**99.97**	6.52/99.76	93.21/94.89	94.14/93.48	93.45/93.50	91.00/94.42	96.72/87.70	**98.31**/38.00	89.23/97.11
Road	0.00/0.00	86.81/**92.48**	98.71/62.71	98.50/60.07	96.65/77.07	98.65/76.78	**99.46**/44.60	98.48/14.34	97.73/86.43
Water	90.99/99.95	90.34/**99.96**	98.67/79.54	99.67/85.96	99.58/94.50	98.73/90.42	99.54/62.69	99.76/0.72	**100.00**/96.79

Table 11: The Precision/Recall for the HanChuan dataset

Class	Class prior-based classifiers		Label noise representation learning				Class prior-free classifiers		
	nnPU	OC Loss	MSE Loss	GCE Loss	SCE Loss	TCE Loss	PAN	vPU	T-HOneCls
Corn	99.89/97.29	99.48/**99.87**	99.96/98.92	99.94/98.38	99.98/97.08	99.96/97.71	99.95/94.59	**100.00**/4.46	99.92/99.49
Sesame	20.00/7.55	61.29/**100.00**	99.93/99.61	**99.97**/99.58	**99.97**/99.59	99.91/99.67	99.94/99.53	**99.97**/53.04	99.94/99.70
Broad-leaf soybean	98.69/74.19	96.10/81.23	**99.98**/69.56	99.92/77.53	99.93/77.20	**99.98**/60.04	**99.98**/41.35	99.93/2.28	99.88/**86.39**
Rice	0.00/0.00	99.41/**100.00**	99.96/97.95	99.96/98.43	99.98/98.36	99.97/97.64	99.99/97.30	**100.00**/21.17	99.96/99.04

Table 12: The Precision/Recall for the LongKou dataset



(a) Positive loss of the variational classifier (b) Total loss of the variational classifier (c) F1-score of the variational classifier

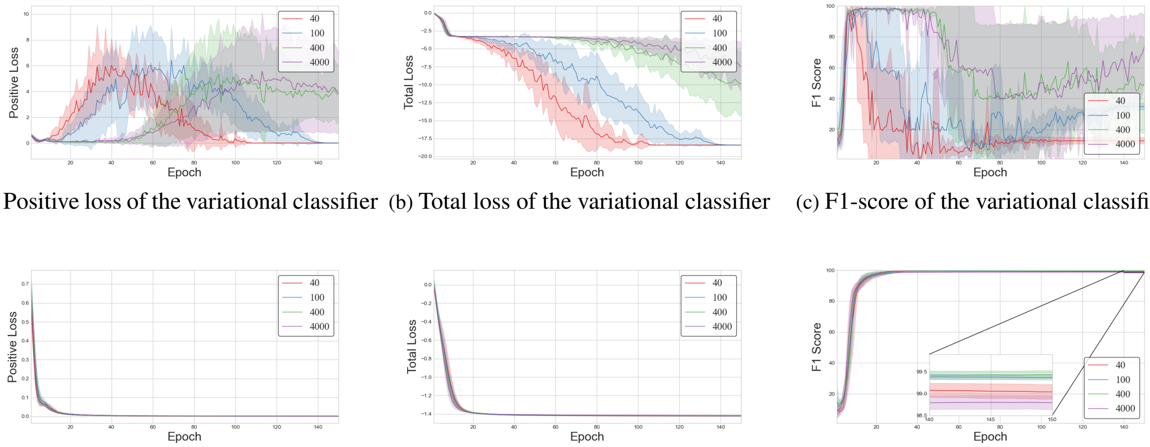
(d) Positive loss of *T-HOneCls* (e) Total loss of *T-HOneCls* (f) F1-score of *T-HOneCls*

Figure 10: The curves of rape in the HongHu dataset, showing the positive loss, total loss, and F1-score of the variational classifier and *T-HOneCls* with different positive training samples in the training stage.

to reduce the weight of the unlabeled data in the gradient. An empirical conclusion can be obtained from Fig. 12: the higher the order of the Taylor expansion, the faster the neural network converges. However, the rapid convergence of the neural network can lead to overfitting. In other words, the classification results first rise and then decline with the progress of the training, as shown in the curve of $o = 5$ in Fig. 12a. A small expansion order slows down the convergence of the neural network, as shown in the curve of $o = 1$ in Fig. 12f. Empirically, a higher Taylor expansion order can be equipped with fewer training epochs, and a smaller Taylor expansion order can be equipped with more training epochs. In order to show that *T-HOneCls* can significantly reduce the overfitting of the neural network for noisy labels, we set a relatively large number of training epochs, so that $o \in \{1, 2, 3, 4\}$ can achieve s good F1-score. Finally, we set $o = 2$.

9. More Experimental Results about KL-Teacher

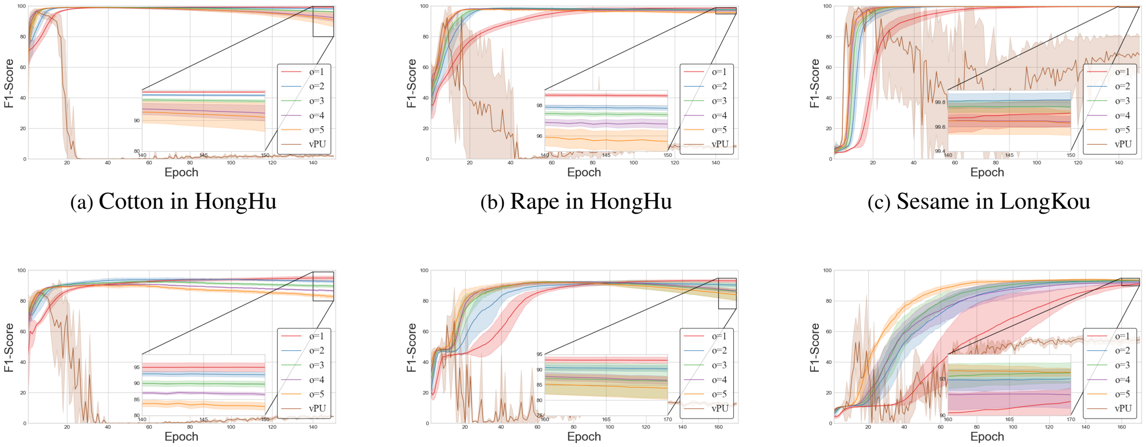
The results of other classes are shown in Fig. 13. The first thing to be analyzed is the role of EMA in the self-calibration optimization. It can be seen from Fig. 13 shows that the F1-score fluctuates greatly when only stochastic gradient descent is used to optimize the *Taylor variational loss*, and in this case, selecting appropriate training epochs can seriously affect the F1-score of the model. EMA has the function of an "F1-score filter", which makes the F1-score of the teacher model more stable, thus reducing the influ-



(a) Positive loss of the variational classifier (b) Total loss of the variational classifier (c) F1-score of the variational classifier

(d) Positive loss of *T-HOneCls* (e) Total loss of *T-HOneCls* (f) F1 score of *T-HOneCls*

Figure 11: The curves of cabbage in the HongHu dataset, showing the positive loss, total loss and F1-score of the variational classifier and *T-HOneCls* with different positive training samples in the training stage.



(a) Cotton in HongHu (b) Rape in HongHu (c) Sesame in LongKou

(d) Broad-leaf soybean in LongKou (e) Cowpea in HanChuan (f) Watermelon in HanChuan

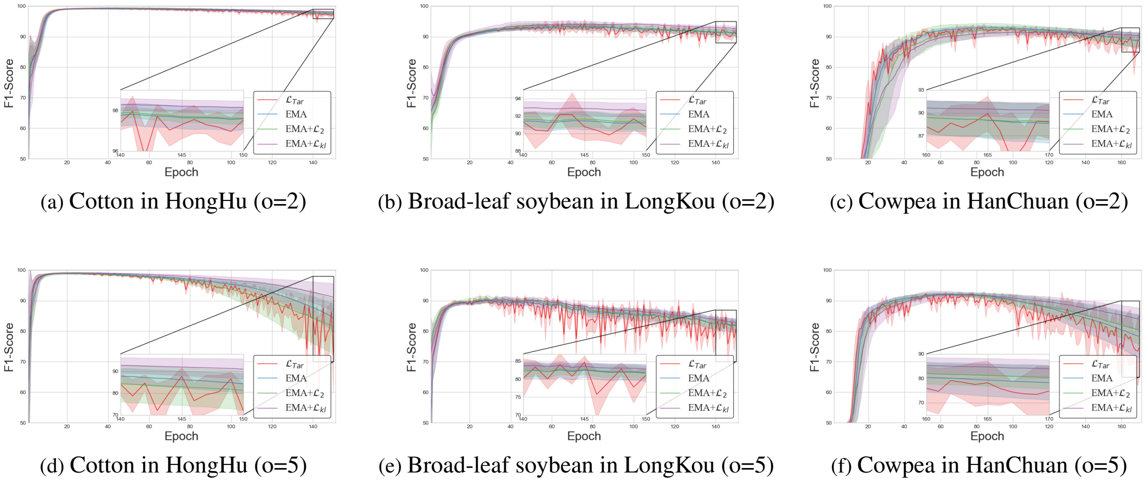
Figure 12: The F1-score curves for the different order of the Taylor series in *T-HOneCls*.

ence of inappropriate training epochs, as shown in Fig. 13.

The exponential moving average allows the teacher model to lag behind the student model, and due to the memorization ability of the neural network, the F1-score of the lagged neural network is better than that of the student network at the later stage of training. The use of consistency loss can promote the output of the student model to approximate the teacher model, so as to alleviate the overfitting problem. If $\mathcal{L}_2$ is regarded as the consistency loss, it is equivalent to Mean-Teacher [34] being used. However, according to the results in Table 6, $\mathcal{L}_2$ cannot effectively alleviate the overfitting of the student model. From Table 6, better F1-score can be obtained by using $\mathcal{L}_{kl}$ as the consistency loss. It can be seen from Fig. 13 that the curve of the F1-score using $\mathcal{L}_{kl}$ is at the top, which indicates that *KL-Teacher* alleviates the overfitting phenomenon to some extent through the memorization ability of the neural network. The analysis of the β in KL-Teacher is presented in the Table 13, and the proposed method is robust to β.

β	0	0.2	0.4	0.5	0.6	0.8	1.0
F1	97.51(0.68)	97.95(0.38)	98.20(0.31)	98.15(0.35)	98.40(0.31)	98.55(0.26)	98.65(0.23)

Table 13: Analysis of the β in the cotton of HongHu dataset.



(a) Cotton in HongHu (o=2) (b) Broad-leaf soybean in LongKou (o=2) (c) Cowpea in HanChuan (o=2)

(d) Cotton in HongHu (o=5) (e) Broad-leaf soybean in LongKou (o=5) (f) Cowpea in HanChuan (o=5)

Figure 13: The F1-score curves of the different components of KL-Teacher in *T-HOneCls*.

10. More Experimental Results about Class Prior-based Method with Oracle Class Prior

The class prior-based method is evaluated with estimated class prior ($\hat{\pi}_p$) and oracle class prior (π_p). Due to the severe inter-class similarity and intra-class variation, the π_p is hard to be estimated accurately in HSI. The π_p and $\hat{\pi}_p$ are shown in the Table 14. The results of class prior-based method are very poor without accurate π_p, the proposed method achieves competitive results compared to the class prior-based method with an oracle π_p (Table 14).

	Class	Rape	Tube mustard	Cowpea	Soybean	Watermelon
Class prior	π_p	0.1317	0.0367	0.0617	0.0279	0.0123
	$\hat{\pi}_p$	0.2231	0.3109	0.2547	0.1509	0.3109
F1-scores	OC Loss(π_p)	**98.73(0.05)**	95.97(0.93)	**90.43(0.48)**	98.04(0.97)	91.21(1.92)
	OC Loss($\hat{\pi}_p$)	81.81(1.23)	23.57(0.22)	58.97(3.56)	42.34(1.06)	12.23(0.46)
	T-HOneCls	97.81(0.16)	**97.38(0.35)**	90.31(1.13)	**99.13(0.28)**	**92.99(0.90)**

Table 14: Comparison of the π_p and the $\hat{\pi}_p$. Comparison of the F1-scores of class prior-based method with π_p and $\hat{\pi}_p$.