LaSSL: Label-guided Self-training for Semi-supervised Learning

Zhen Zhao[1], Luping Zhou[1*], Lei Wang[2], Yinghuan Shi[3*], Yang Gao[3]
[1] School of Electrical and Information Engineering, University of Sydney, Australia
[2] School of Computing and Information Technology, University of Wollongong, Australia
[3] National Key Laboratory for Novel Software Technology, Nanjing University, China
{zhen.zhao, luping.zhou}@sydney.edu.au, leiw@uow.edu.au, {syh,gaoy}@nju.edu.cn

（AAAI-2022)

code : https://github.com/zhenzhao/lassl

gesis
Leibniz-Institut für Sozialwissenschaften

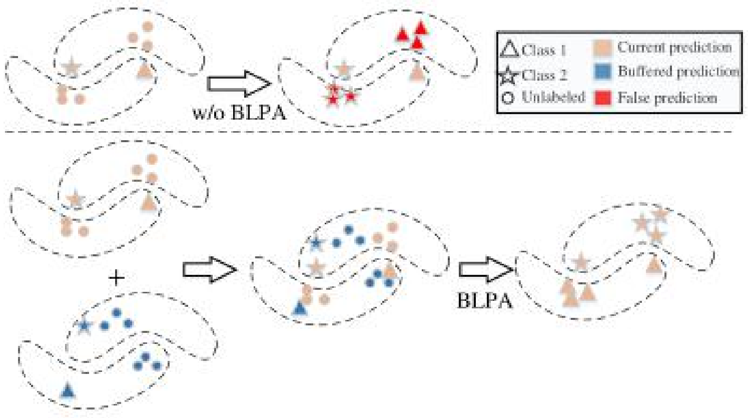
Reported by Zhaoze Gao

1. Introduction
2. Approach
3. Experiments

Introduction

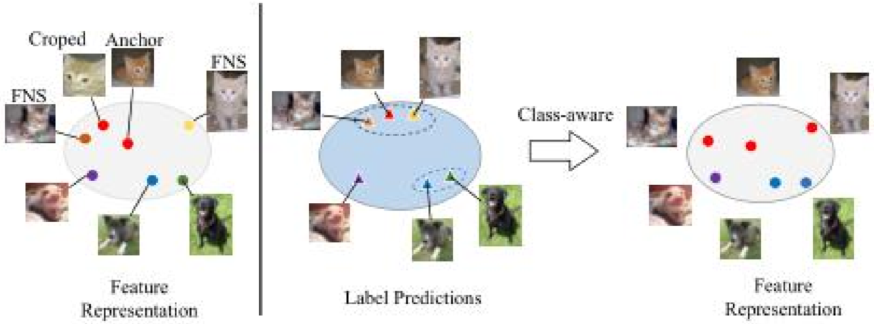
Buffer-aided Label Propagation Algorithm

we propagate the labels from the labeled samples to the unlabeled ones across the underlying data manifold via the label propagation algorithm (LPA) at the feature-embedding level.we could take advantage of the correlation between the labeled and unlabeled samples to improve pseudo-label generation.



(a) BLPA

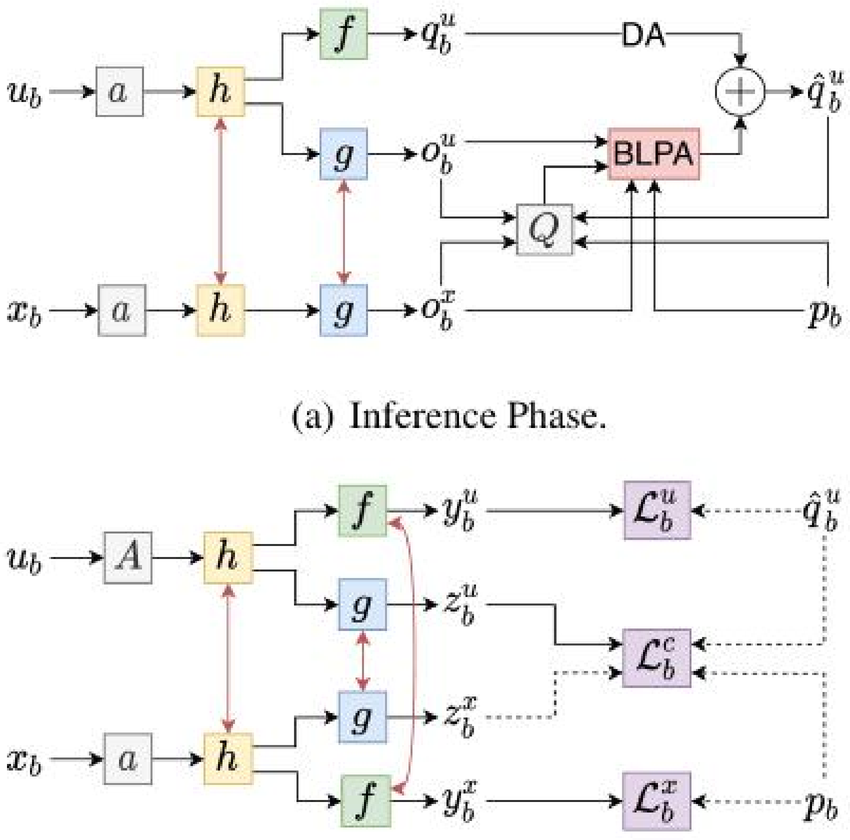
LaSSL explores a better feature embedding through a proposed class-aware contrastive loss, so that the same-class samples are gathered and the different-class samples are scattered.



(b) CACL

Class-aware Contrastive Loss

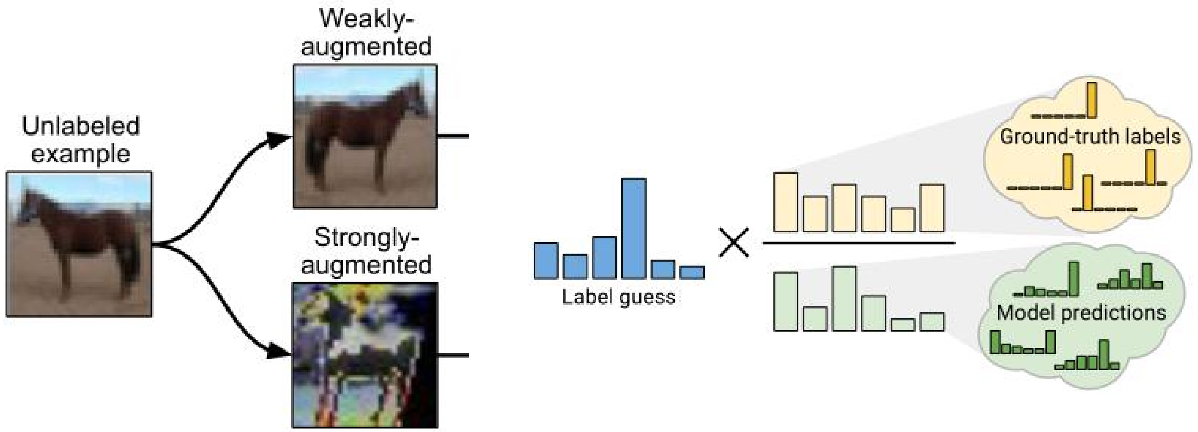
Approach



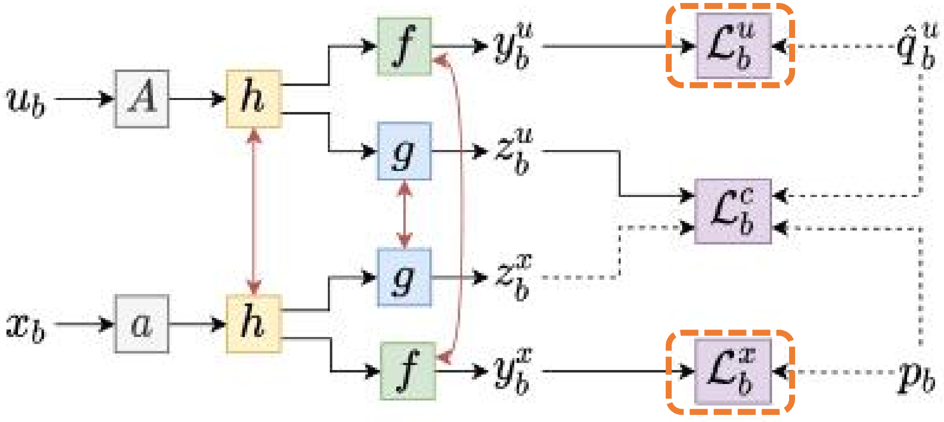
(a) Inference Phase.

(b) Training Phase.

a weak augmentations
A strong augmentations
h encoder
f predictor
g projector to learn feature representations.
DA(distribution alignment)
Q:queue



Approach



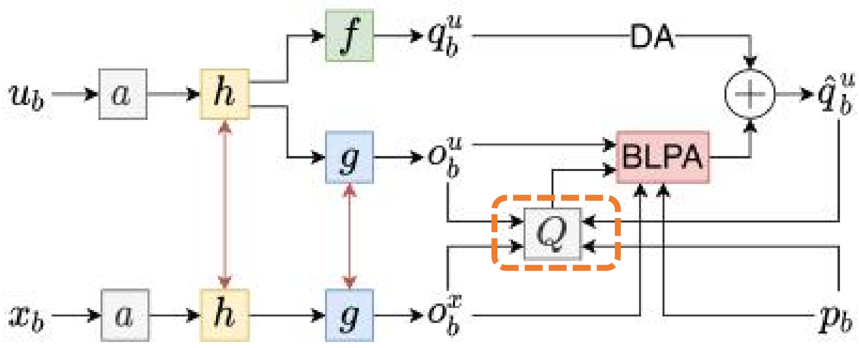
(b) Training Phase.

Let (x_b, p_b) be a batch of B labeled samples and u_b be a batch of μB unlabeled samples where μ denotes the size ratio of x_b to u_b.

$$\mathcal{L}_b^x = H(p_b, y_b^x) \quad (1)$$
$$\mathcal{L}_b^u = \mathbf{1}(\max(\hat{q}_b^u) \geq \tau)\, H(\hat{q}_b^u, y_b^u) \quad (2)$$

where $\mathbf{1}(\cdot)$ retains the pseudo-labels whose maximum probability is higher than a predefined threshold τ, i.e. high-

Approach



(a) Inference Phase.

$$Q_i = \{(o_b, q_b)\} \qquad o_b \in \{o_b^u\} \cup \{o_b^x\},\ q_b \in \{\hat{q}_b^u\} \cup \{p_b\}$$

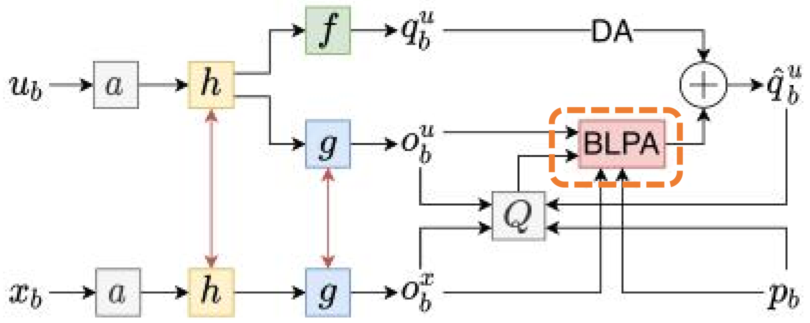
high-confidence labels can inevitably include errors. In order to decrease the noise, we do K random sampling with replacement on the dequeue data (i.e. bagging), and denote each sampling result as $o_{b-1}(k)$ and $q_{b-1}(k)$, where $k = 1, 2, \ldots K$. After that, we can split the sampling data with

$$q_{b-1}^{high}(k) = \mathbf{1}(\max(q_{b-1}(k)) \geq \tau)\, q_{b-1}(k), \tag{3}$$

$$o_{b-1}^{high}(k) = \mathbf{1}(\max(q_{b-1}(k)) \geq \tau)\, o_{b-1}(k), \tag{4}$$

$$o_{b-1}^{low}(k) = \mathbf{1}(\max(q_{b-1}(k)) < \tau)\, o_{b-1}(k). \tag{5}$$

Approach



(a) Inference Phase.

$$o_s(k) = [o_b^x, o_{b-1}^{high}(k)], \qquad o_t(k) = [o_b^u, o_{b-1}^{low}(k)],$$

$$q_s(k) = [p_b, q_{b-1}^{high}(k)].$$

$$\widetilde{\Omega}(k) = D^{-1/2}\Omega(k)D^{1/2} \tag{6}$$

$$\Phi_{j+1}(k) = \alpha\widetilde{\Omega}(k)\Phi_j(k) + (1-\alpha)q_s(k) \tag{7}$$

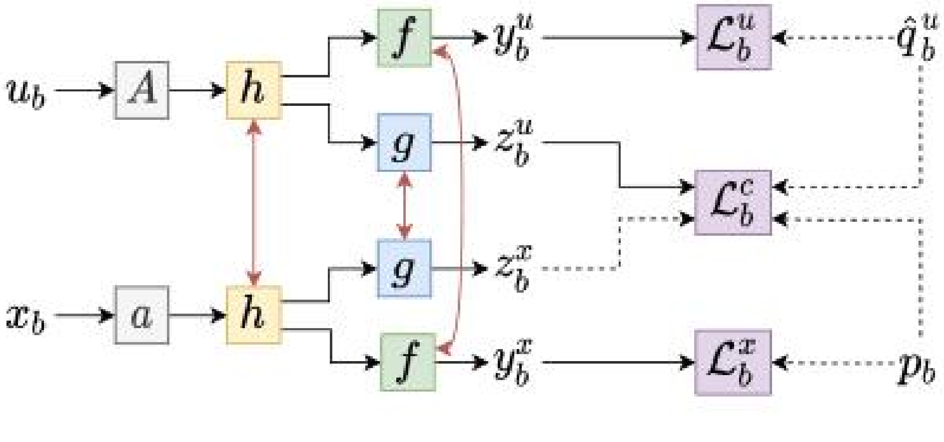
$$\Phi^*(k) = (I - \alpha\widetilde{\Omega}(k))^{-1}q_s(k). \tag{8}$$

$\phi_b(k) = \Phi^*(k)[:\mu B]$. μ denotes the size ratio of x_b to u_b.

$$\widetilde{q}_b^u = \frac{1}{K}\sum_{k=1}^{K}\phi_b(k). \tag{9}$$

$$\hat{q}_b^u = \eta\widetilde{q}_b^u + (1-\eta)\bar{q}_b^u, \tag{10}$$

Approach



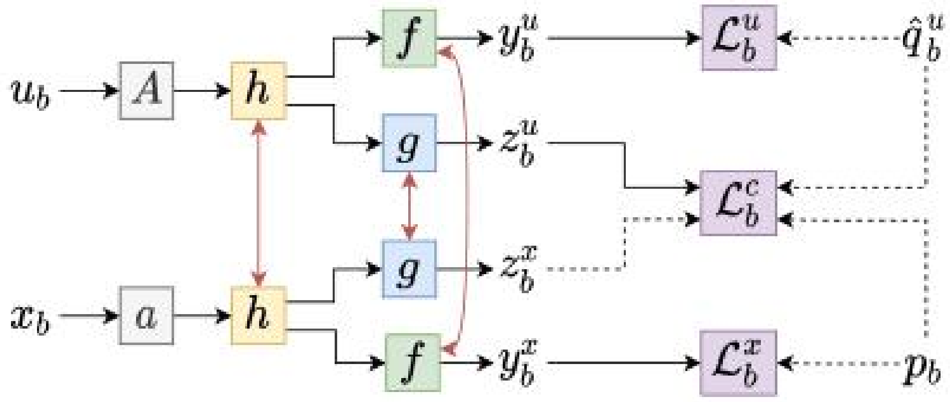
(b) Training Phase.

$$\hat{y} = [p_b, \hat{q}_b^u]$$

$$\omega_{i,j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \text{ and } \hat{y}_i \cdot \hat{y}_j < \varepsilon \\ \hat{y}_i \cdot \hat{y}_j, & \text{if } i \neq j \text{ and } \hat{y}_i \cdot \hat{y}_j \geq \varepsilon \end{cases} \tag{11}$$

$$\mathcal{L}_b^c = -\sum_{i=1}^{|\hat{y}|} \log \frac{\sum_{j=1}^{|\hat{y}|} \omega_{i,j} \exp(z_i \cdot z_j / T)}{\sum_{j=1, j \neq i}^{|\hat{y}|} \exp(z_i \cdot z_j / T)}. \tag{12}$$

Approach



(b) Training Phase.

$$\mathcal{L}_b = \mathcal{L}_b^x + \lambda_u \mathcal{L}_b^u + \lambda_c \mathcal{L}_b^c, \tag{13}$$

$$\lambda_c = \begin{cases} \lambda_c^0, & \text{if } t \leq T_r, \\ \lambda_c^0 \exp\left(-\dfrac{(t - T_r)^2}{2(T_t - T_r)}\right), & \text{otherwise.} \end{cases} \tag{14}$$

Experiments

Methods	CIFAR-10 40 labels	CIFAR-10 250 labels	CIFAR-100 400 labels	CIFAR-100 2500 labels	SVHN 40 labels	SVHN 250 labels
Π-Model*	-	45.74±3.87	-	42.75±0.48	-	81.04±1.92
Pseudo-label*	-	50.22±0.43	-	42.62±0.46	-	79.79±1.09
Mean-Teacher*	-	67.68±2.30	-	46.09±0.57	-	96.43±0.11
MixMatch*	52.46±11.50	88.95±0.86	33.39±1.32	60.06±0.37	57.45±14.53	96.02±0.23
UDA*	70.95±5.93	91.18±1.08	40.72±0.88	66.87±0.22	47.37±20.51	94.31±2.76
ReMixMatch*	80.90±9.64	94.56±0.05	55.72±2.06	72.57±0.31	96.64±0.30	97.08±0.48
FixMatch*	86.19±3.37	94.93±0.65	51.15±1.75	71.71±0.11	96.04±2.17	97.52±0.38
ACR†	92.38	95.01	-	-	-	-
SelfMatch†	93.19±1.08	95.13±0.26	-	-	96.58±1.02	97.37±0.43
CoMatch†	93.09±1.39	95.09±0.33	-	-	-	-
Dash†	86.78±3.75	95.44±0.13	55.24±0.96	72.82±0.21	**96.97±1.59**	97.83±0.10
LaSSL	**95.07± 0.78**	**95.71 ±0.46**	**62.33±2.69,**	**74.67± 0.65**	96.91±0.52	**97.85± 0.13**

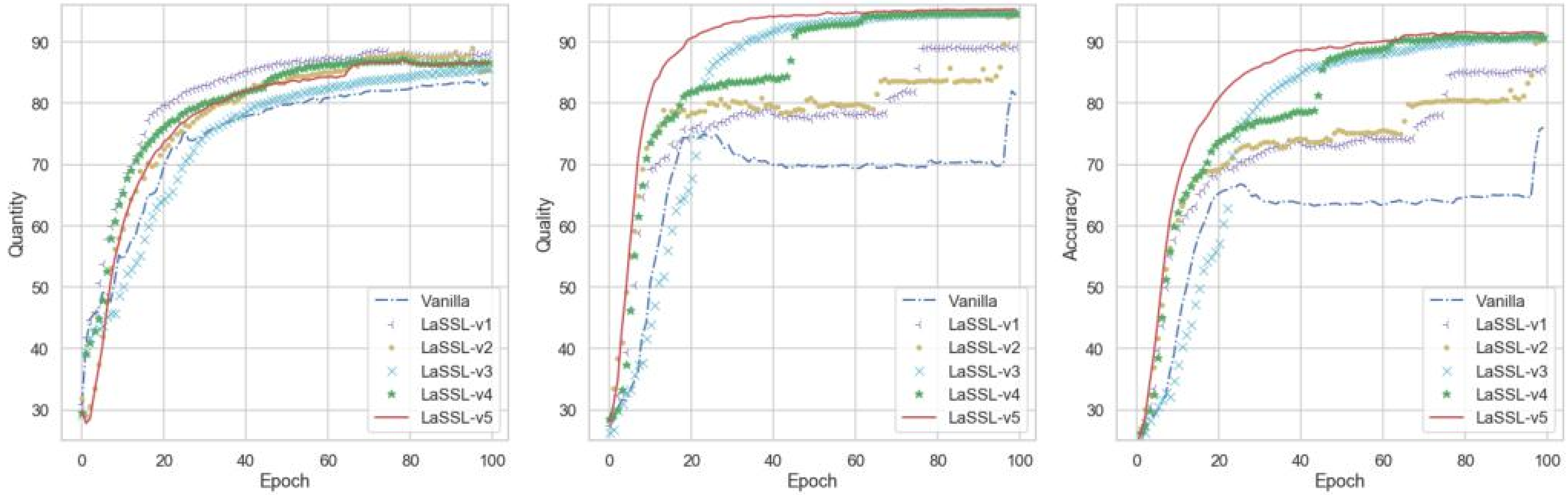
Table 1: Top-1 testing accuracy (%) for CIFAR-10, CIFAR-100 and SVHN on 5 different folds. All the related works are sorted by their publication date. Results with * was reported in FixMatch (Sohn et al. 2020), while results with † comes from the most recent papers (Kim et al. 2021; Li, Xiong, and Hoi 2020; Xu et al. 2021; Abuduweili et al. 2021), respectively.

Experiments

Method	CACL	BLPA	DA	Quantity	Quality	Accuracy
Vanilla	✗	✗	✗	83.91	81.98	75.54
LaSSL-v1	✓	✗	✗	88.66	89.38	85.50
LaSSL-v2	✓	✓	✗	89.08	94.31	90.24
LaSSL-v3	✗	✗	✓	85.73	94.90	90.42
LaSSL-v4	✓	✗	✓	87.46	94.89	91.11
LaSSL-v5	✓	✓	✓	87.03	**95.33**	**91.65**

Table 2: Ablation studies on CIFAR-10 with 40 labeled data after training 100 epochs (random seed is fixed to 1.)

Experiments



(a) quantity (b) quality (c) accuracy

Figure 3: (a), (b), (c) represent curves of the quantity, quality, and EMA test accuracy of different combinations of CACL, BLPA, and DA (better view on screen). Numerical results are listed in Table 2.

Experiments

ε	0.6	0.7	0.8	0.9	1.0
Accuracy(%)	87.64	**89.39**	87.70	87.36	85.17

Table 3: Effects with different similarity thresholds. The similarity is equal to 1 only when comparing the image instance with itself. Therefore, we use $\varepsilon = 1.0$ to investigate the effect of excluding the "class-aware" technique.

Experiments

K	0	1	3	5	7
Accuracy(%)	92.71	92.10	94.64	93.43	**94.87**

Table 4: Effects with different number of samplings. In specific, $K = 0$ means the plain LPA without "buffer-aided"; $K = 1$ means exploiting the buffered data directly without sampling; while $K > 1$ investigates the complete BLPA.

Thank you !