Sli2Vol: Annotate a 3D Volume from a Single Slice with Self-Supervised Learning

MICCAI 2021

Motivation

1. Annotations for the training volumes are usually a **costly** process to acquire.
2. Once **domain shift** appears, the model may suffer a catastrophic drop in performance.

Contribution

1. A **self-supervised training** semi-automatic segmentation for 3D volume with **single slice annotation** during **inference**.
2. Propose a mask **propagation** approaches based on learning to **match slices' correspondences** and using a newly proposed **edge profile** for information bottleneck.
3. Propose and exploit a simple **verification module** for refining the mask during inference time to alleviate the **error accumulation** in mask propagation.

Framework

[Self-supervised Learning for Video Correspondence Flow]

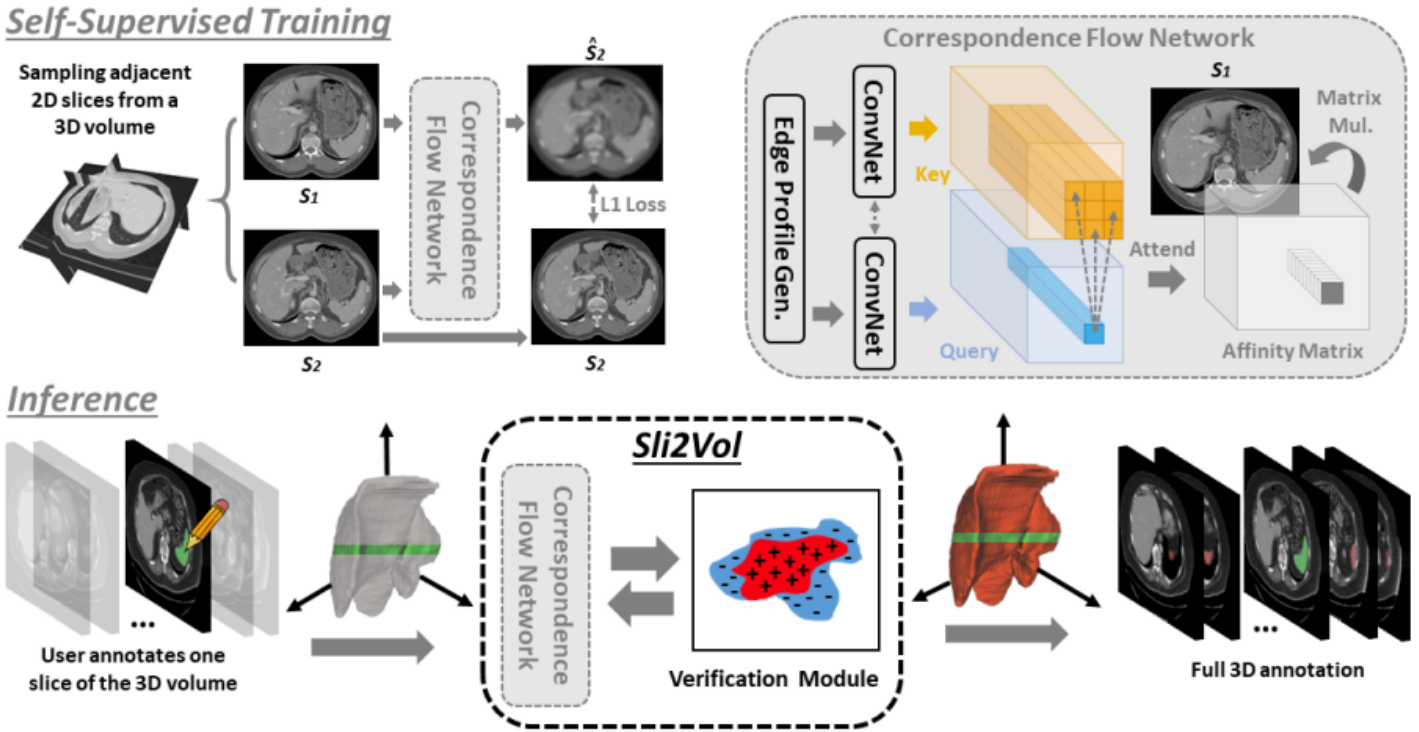


Fig. 1. Pipeline of our proposed framework. During *self-supervised training*, pair of adjacent slices sampled from 3D volumes are used to train a correspondence flow network. Provided with the 2D mask of a single slice of a volume, the trained network with the verification module can be used to propagate the initial annotation to the whole volume during *inference*.

Method: Self-Supervised Training of Sli2Vol

- The idea is to task a deep network for slice reconstruction by weighting and copying pixels from its neighboring slice.

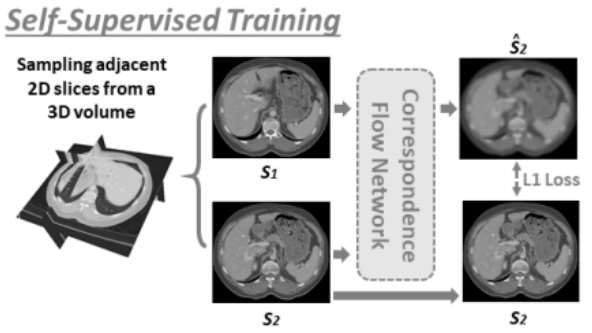
During training, a pair of adjacent slices, $\{\mathbf{S}_1, \mathbf{S}_2\}, \mathbf{S}_i \in \mathcal{R}^{H \times W \times 1}$

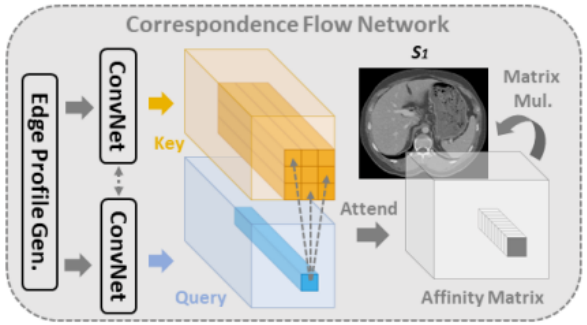
$$[\mathbf{k}_1,\ \mathbf{q}_2] = [\psi(g(\mathbf{S}_1);\ \theta),\ \psi(g(\mathbf{S}_2);\ \theta)]$$

[$g(\cdot)$ denotes an *edge profile generator*]

$$\mathbf{A}_{1\to 2}(u,v) = \frac{exp\langle \mathbf{q}_2(u,:), \mathbf{k}_1(v,:)\rangle}{\sum_{\lambda\in\Omega} exp\langle \mathbf{q}_2(u,:), \mathbf{k}_1(\lambda,:)\rangle}$$

$$\hat{\mathbf{S}}_2(u,1) = \sum_{v}^{\Omega} \mathbf{A}_{1\to 2}(u,v)\mathbf{S}_1(v,1).$$





Method: Edge Profile Generator

- Question
  - The model (CorrFlow) must learn to establish reliable correspondences between the two slices
  - But that may actually incur **trivial solutions**, simply matching the pixel intensity of S1 and S2
- Previous Solution
  - information bottleneck: input **color channel** (i.e. RGB or Lab) **dropout**
  - ...which breaks the correlation between the color channels and forces the model to learn more robust correspondences
  - not feasible in medical images
- Proposed
  - information bottleneck: edge profile
  - For **each pixel**, we convert its **intensity value** to a normalized edge histogram, by computing the derivatives along **d different directions at s different scales**, i.e. $g(\mathbf{S}_i) \in \mathcal{R}^{H \times W \times (d \times s)}$, followed by a **softmax** normalization through all the derivatives
  - Intuitively, g(·) explicitly represents the **edge distributions** centered each pixel of the slice Si , and force the model to pay **more attentions to the edges** during reconstruction.

Method: Verification Module

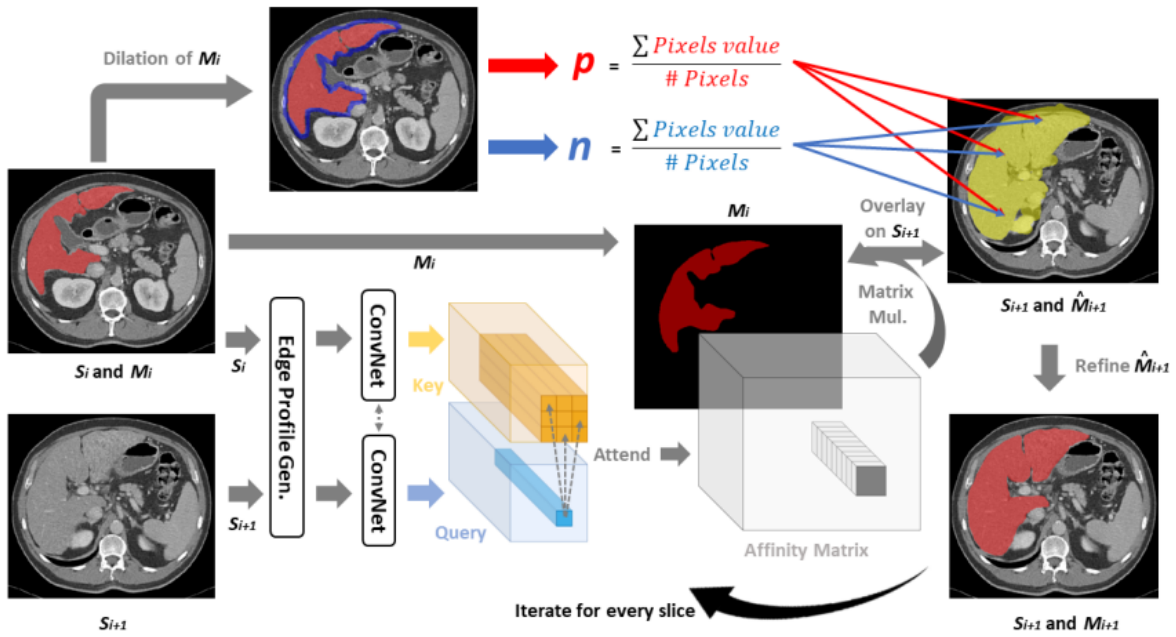


Fig. 2. Computation of each iteration of **Sli2Vol** during *inference*. $\{\mathbf{S}_i, \mathbf{S}_{i+1}\}$, sampled from $\mathbf{V}$ are fed into the trained correspondence flow network to obtain the affinity matrix to propagate $\mathbf{M}_i$ to $\hat{\mathbf{M}}_{i+1}$. $\hat{\mathbf{M}}_{i+1}$ is then refined by $\mathbf{p}$ and $\mathbf{n}$, obtained by $\mathbf{M}_i$ and $\mathbf{S}_i$, to get the final mask, $\mathbf{M}_{i+1}$.

..two regions, namely positive $\mathbf{P} \in \mathcal{R}^{H \times W}$) and negative ($\mathbf{N} \in \mathcal{R}^{H \times W}$) regions, are constructed.

We maintain the mean intensity value within each region:

$$p = \frac{1}{|P_i|}\langle P_i, S_i \rangle \qquad n = \frac{1}{|N_i|}\langle N_i, S_i \rangle$$

After propagation, predicted mask $\hat{\mathbf{M}}^u_{i+1}$ compared to p and n and being re-classified

$$\mathbf{M}^u_{i+1} = \begin{cases} 1, & \text{if } \hat{\mathbf{M}}^u_{i+1} = 1 \text{ and } \sqrt{(\mathbf{S}^u_{i+1} - p)^2} < \sqrt{(\mathbf{S}^u_{i+1} - n)^2} \\ 0, & \text{otherwise} \end{cases}$$

Experiment

- For **chest and abdominal CT**, a single model is **trained on 3 unannotated dataset** (i.e. C4KC-KiTS, CTLN and CT-Pancreas) and **tested on 7 other datasets** (i.e. Sliver07, CHAOS, 3Dircadb-01, 02, and Decath-Spleen, Liver and Pancreas)
- For **cardiac MRI**, models are **trained on the 2D video dataset** from Kaggle, and **tested on a 3D volume dataset** (i.e. Decath-Heart), which manifests large domain shift.
- **FS - Single Slice**: For example, in Sliver07, the model trained on 20 slice annotations (single slice from each volume), is tested on the **same set** of 20 volumes.
- For **Sli2Vol, FS - Single Slice, Optical Flow and VM**, we randomly pick one of the ±3 slices around the slice with the **largest ground-truth annotation** as the initial mask.
  - This simulates the process of a user sliding through the whole volume and roughly identifying the slice with the largest SOI to annotate, which is achievable in reality.

Modality	*MRI*	*Abdominal and Chest CT*													
Training Dataset (for row e to j)	*Kaggle*	*C4KC-KiTS, CT-LN and CT-Pancreas*													
Testing Dataset	*Decath -Heart*	*Sliver07*	*CHAOS*	*Decath -Liver*	*Decath -Spleen*	*Decath-Pancreas*	*3D-IRCADb-01 and 3D-IRCADb-02*								
ROI	*Left Atrium*	*Liver*	*Liver*	*Liver*	*Spleen*	*Pancreas*	*Heart*	*Gall-bladder*	*Kidney*	*Surrenal-gland*	*Liver*	*Lung*	*Pancreas*	*Spleen*	*Mean Results*
Number of Volumes	*20*	*20*	*20*	*131*	*41*	*281*	*3*	*8*	*17*	*11*	*22*	*12*	*4*	*7*	
Automatic (Trained with Fully Annotated Data)															
(a) Fully Supervised-same domain	92.7 [11]	94.8 [2] (93.9)	97.8 [13] (92.8)	95.4 [11] (91.0)	96.0 [11]	79.3 [11]	-	-	-	-	96.5 [24]	-	-	-	-
(b) Fully Supervised-different domain	-	74.8 ±13.2	76.5 ±8.8	56.0 ±23.6	-	-	-	-	-	-	-	-	-	-	-
Semi-automatic															
(c) Fully Supervised-single slice	62.5 ±5.2	86.9 ±4.1	84.3 ±4.1	85.0 ±5.5	74.4 ±12.0	49.9 ±13.4	25.6 ±6.5	47.9 ±15.5	57.9 ±21.1	30.8 ±15.6	80.3 ±13.8	81.0 ±10.8	20.4 ±7.9	58.6 ±4.7	60.4
(d) Optical Flow	51.1 ±7.4	65.2 ±8.8	72.0 ±9.9	47.0 ±15.9	72.9 ±14.5	25.1 ±8.2	32.2 ±11.6	24.6 ±12.4	73.6 ±14.6	22.1 ±12.9	68.4 ±9.4	33.6 ±18.0	21.9 ±12.6	70.8 ±17.5	48.6
(e) VoxelMorph2D-UNet	42.9 ±5.0	57.2 ±9.8	66.5 ±10.5	38.5 ±12.5	61.5 ±19.5	21.4 ±6.7	20.3 ±6.5	20.2 ±12.2	70.1 ±18.6	41.1 ±15.3	60.5 ±9.7	38.7 ±21.2	28.3 ±11.0	54.1 ±12.4	44.4
(f) VoxelMorph2D-ResNet18NoStride	45.7 ±4.1	61.2 ±8.5	68.4 ±9.8	42.2 ±12.4	58.3 ±17.3	23.5 ±7.8	22.1 ±6.7	21.8 ±13.1	77.8 ±18.4	48.4 ±15.3	60.6 ±10.4	36.5 ±20.0	32.3 ±13.3	60.0 ±12.1	47.5
Sli2Vol	*Ablation Studies*														
(g) Correspondence Flow Network	62.4 ±9.2	75.0 ±6.5	78.9 ±7.9	66.0 ±13.1	81.1 ±13.9	43.9 ±12.9	55.4 ±24.3	62.4 ±20.7	86.0 ±19.0	45.9 ±18.6	75.0 ±8.6	45.2 ±25.4	44.3 ±17.2	81.8 ±19.6	64.5
(h) Network + Edge Profile	56.8 ±8.4	74.8 ±7.4	77.8 ±8.4	64.4 ±14.1	83.6 ±13.2	48.9 ±11.2	49.4 ±12.3	68.5 ±13.8	86.8 ±15.7	**58.3 ±16.6**	73.9 ±8.5	48.8 ±26.4	53.9 ±7.1	85.8 ±13.0	66.6
(i) Network + Verif. Module	**80.8 ±5.0**	81.1 ±5.0	83.4 ±6.3	72.0 ±8.9	79.1 ±17.3	37.3 ±13.6	50.9 ±11.6	70.7 ±12.7	83.3 ±21.4	47.5 ±20.8	78.8 ±6.9	79.8 ±29.3	45.2 ±10.5	74.5 ±23.7	68.9
(j) Network + Verif. Module + Edge Profile	80.4 ±4.5	**91.3 ±3.2**	**91.0 ±2.9**	**86.8 ±7.2**	**88.4 ±10.9**	**54.2 ±10.0**	**75.9 ±10.9**	**68.9 ±9.9**	**91.4 ±4.8**	48.4 ±13.5	**88.2 ±3.0**	**81.4 ±28.5**	**58.2 ±4.6**	**90.2 ±9.5**	**78.2**

Table 1. Results (mean Dice scores ± standard deviation) of different approaches on different datasets and SOIs. Higher value represents better performance. In **row a**, results from both state-of-the-art methods [2,11,13] and 3D UNets trained by us (values in the bracket) are reported. Results in **row a** and **b** are only partially available in literature and they are reported just for demonstrating the approximated upper bound and limitation of fully supervised approaches, which are not meant to be directly compared to our proposed approach.

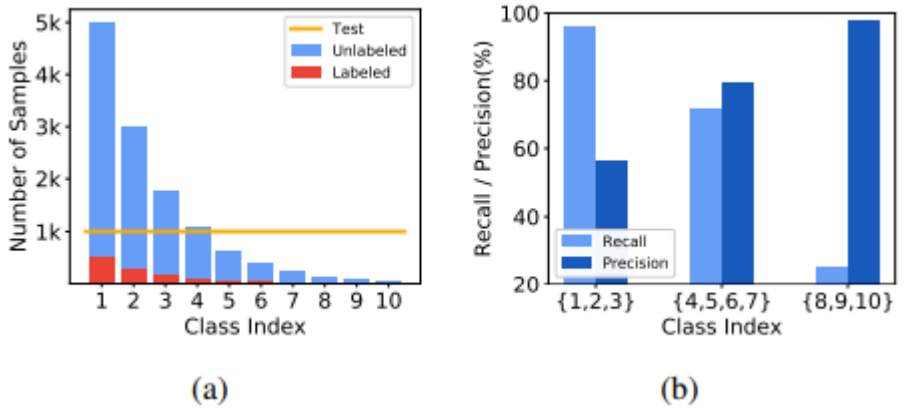
The End

CReST: A Class-Rebalancing Self-Training Framework for Imbalanced Semi-Supervised Learning

CVPR 2021

Motivation

- Various solutions have been proposed to help alleviate bias (from imlablanced data), such as **re-sampling**, **re-weighting**, and **two-stage training**, but SSL on imbalanced data has been **understudied**.
- Pseudo-labels can be **problematic** if they are generated by an initial model trained on **imbalanced data** and biased toward majority classes
- While existing semi-supervised learning (SSL) methods are known to perform poorly on minority classes, we find that they still generate **high precision** pseudo-labels on **minority** classes.



Experimental results on CIFAR10-LT.
(a) Both labeled and unlabeled sets are class-imbalanced, where the most majority class has 100× more samples than the most minority class. The test set remains balanced.
(b) Precision and recall of a FixMatch model.

Closer look: model bias

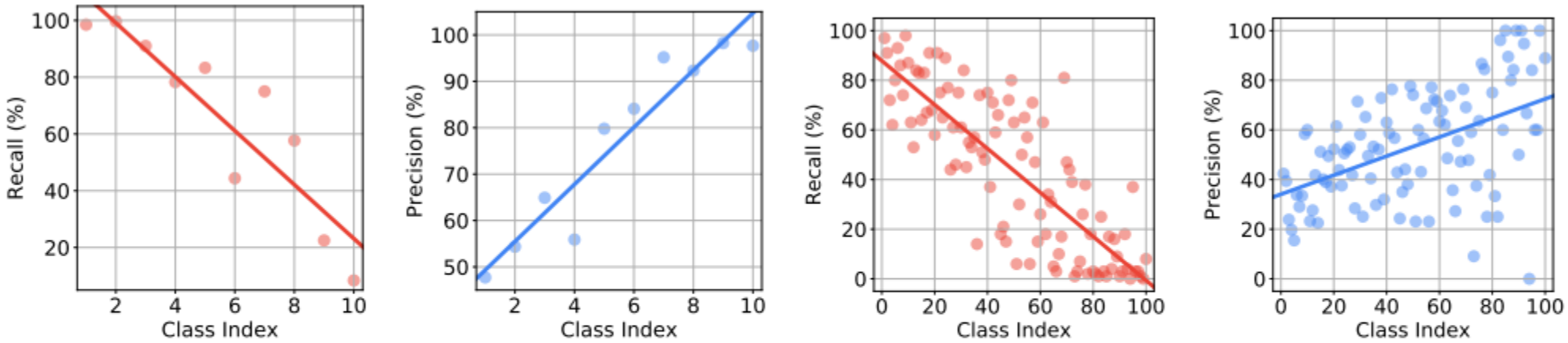


Figure 2. Bias of a FixMatch [39] model on class-imbalanced data. **Left**: Per-class recall and precision on CIFAR10-LT. **Right**: Per-class recall and precision on CIFAR100-LT. The class index is sorted by the number of examples in descending order. While the conventional assumption might be that the performance of the majority classes is better than that of the minority classes, we find it only partially true. The model obtains high recall but low precision on majority classes, while obtaining low recall but high precision on minority classes. See more details in Sec. 3.2.

the model has almost perfect precision on minority classes, suggesting that the model is **conservative** in classifying samples into **minority** classes,

CReST: Class-rebalancing self-training

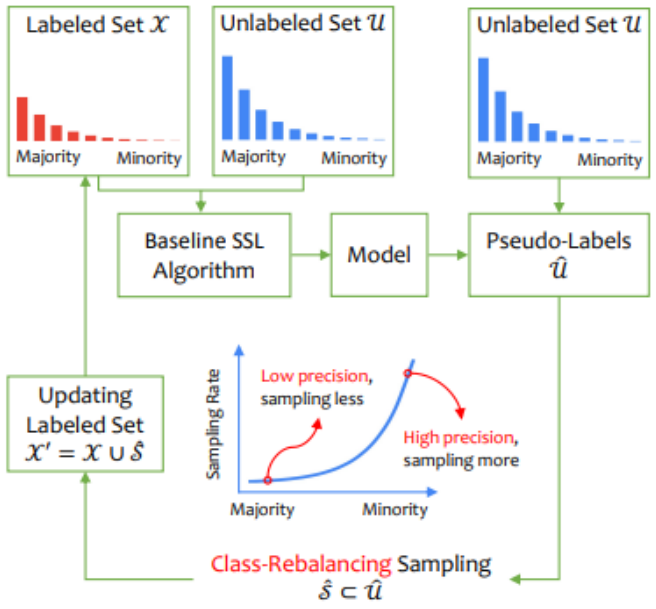


Figure 3. CReST (Class-Rebalancing Self-Training) alternatingly trains a baseline SSL algorithm on both labeled and unlabeled data and expands the labeled set by sampling pseudo-labeled unlabeled data. Sampling rates for majority and minority classes are adaptively determined based on the quality of pseudo-labels. See text for details.

Sampling strategy:
the **less frequent** a class l is, the **more unlabeled samples that are predicted** as class l are included into the pseudo-labeled set $\hat{\mathcal{S}}$

Sampling rate:
1. estimate the **class distribution** from the labeled set.
2. unlabeled samples that are predicted as class l are included into Ŝ a

$$\mu_l = \left(\frac{N_{L+1-l}}{N_1}\right)^{\alpha}$$

For instance, for a 10-class imbalanced dataset with imbalance ratio of γ = **N1/N10 = 100**, we keep all samples predicted as the most minority class since ***μ*10** = ((N10+1−10)/N1)^α = **1**. While for the most majority class, ***μ*1** = ((N10+1−1)/N1)^α = **0.01α** of samples are selected.

CReST+: Progressive distribution alignment

- DA(Distribution Alignment) [ReMixMatch]
  - It **aligns** the model's predictive distribution on unlabeled samples with the labeled training set's class distribution p(y).
- Smoother DA
  - To further enhance DA's ability to handle class imbalanced data, we extend it with **temperature scaling t**.
  - t=1, recover DA; t<1, become smoother; t=0, become uniform
- Progressive DA
  - Over-balanced: more samples are wrongly predicted as minority classes
  - Propose to progressively increase the strength of class-rebalancing by **decreasing t** over generations.

$$t_g = (1 - \frac{g}{G}) \cdot 1.0 + \frac{g}{G} \cdot t_{\min}$$

g: current generation
G: G + 1 is the total number of generations
t_min: the temperature used for the last generation

  - This progressive schedule for t enjoys both **high precision** of pseudo labels in **early generations**, and stronger **class-rebalancing** in **late generations**.

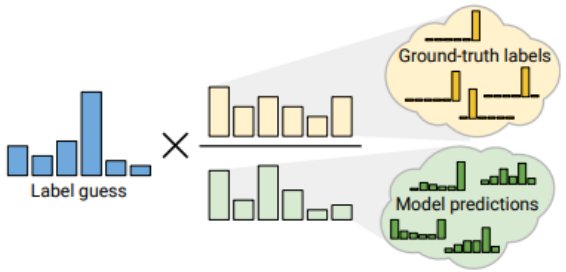


Figure 1: Distribution alignment. Guessed label distributions are adjusted according to the ratio of the empirical ground-truth class distribution divided by the average model predictions on unlabeled data.

Experiment-Comparison with baseline

	CIFAR10-LT						CIFAR100-LT			
	$\beta = 10\%$			$\beta = 30\%$			$\beta = 10\%$		$\beta = 30\%$	
Method	$\gamma = 50$	$\gamma = 100$	$\gamma = 200$	$\gamma = 50$	$\gamma = 100$	$\gamma = 200$	$\gamma = 50$	$\gamma = 100$	$\gamma = 50$	$\gamma = 100$
FixMatch [39]	$79.4_{\pm 0.65}$	$66.3_{\pm 1.74}$	$59.7_{\pm 0.74}$	$81.9_{\pm 0.30}$	$73.1_{\pm 0.58}$	$64.7_{\pm 0.69}$	$33.7_{\pm 0.94}$	$28.3_{\pm 0.66}$	$43.1_{\pm 0.24}$	$38.6_{\pm 0.45}$
w/ CReST	$83.8_{\pm 0.45}$	$75.9_{\pm 0.62}$	$64.1_{\pm 0.23}$	$84.2_{\pm 0.13}$	$77.6_{\pm 0.86}$	$67.7_{\pm 0.82}$	$37.4_{\pm 0.29}$	$32.1_{\pm 1.52}$	$45.6_{\pm 0.19}$	$40.2_{\pm 0.53}$
w/ CReST+	$\mathbf{84.2}_{\pm 0.39}$	$\mathbf{78.1}_{\pm 0.84}$	$\mathbf{67.7}_{\pm 1.39}$	$\mathbf{84.9}_{\pm 0.27}$	$\mathbf{79.2}_{\pm 0.20}$	$\mathbf{70.5}_{\pm 0.56}$	$\mathbf{38.8}_{\pm 1.03}$	$\mathbf{34.6}_{\pm 0.74}$	$\mathbf{46.7}_{\pm 0.34}$	$\mathbf{42.0}_{\pm 0.44}$

Table 1. Classification accuracy (%) on CIFAR10-LT and CIFAR100-LT under various label fraction β and imbalance ratio γ. The numbers are averaged over 5 different folds. Models with CReST are trained for 15 generations. Models with CReST+ are trained for 6 generations.

Experiment-Comparison & Ablasion

Method	$\gamma = 50$	$\gamma = 100$	$\gamma = 200$
Pseudo-Labeling [26]	$52.5_{\pm 0.74}$	$46.5_{\pm 1.29}$	$42.0_{\pm 1.39}$
Mean Teacher [43]	$57.1_{\pm 3.00}$	$48.1_{\pm 0.71}$	$45.1_{\pm 1.28}$
MixMatch [2]	$69.1_{\pm 1.18}$	$60.4_{\pm 2.24}$	$54.5_{\pm 1.87}$
w/ CReST	$69.8_{\pm 1.06}$	$60.5_{\pm 1.56}$	$55.2_{\pm 2.25}$
w/ CReST+	$76.7_{\pm 0.35}$	$66.1_{\pm 0.79}$	$57.6_{\pm 1.30}$
FixMatch [39]	$80.1_{\pm 0.44}$	$67.3_{\pm 1.19}$	$59.7_{\pm 0.63}$
w/ CB [9]	$80.2_{\pm 0.45}$	$67.6_{\pm 1.88}$	$60.8_{\pm 0.26}$
w/ RS [3, 4]	$80.2_{\pm 0.78}$	$69.6_{\pm 1.30}$	$60.9_{\pm 1.25}$
w/ DA [1] ($\iota = 1.0$)	$80.2_{\pm 0.45}$	$69.7_{\pm 1.27}$	$62.0_{\pm 0.84}$
w/ DA [1] ($t = 0.5$)	$82.4_{\pm 0.33}$	$73.6_{\pm 0.63}$	$63.7_{\pm 1.17}$
w/ LA [31]	$83.2_{\pm 0.87}$	$70.4_{\pm 2.90}$	$62.4_{\pm 1.24}$
w/ CReST	$83.2_{\pm 0.37}$	$74.8_{\pm 1.09}$	$63.4_{\pm 0.32}$
w/ CReST+	$84.2_{\pm 0.39}$	$78.1_{\pm 0.84}$	$67.7_{\pm 1.39}$
w/ CReST+ & LA	$\mathbf{85.6}_{\pm 0.36}$	$\mathbf{81.2}_{\pm 0.70}$	$\mathbf{71.9}_{\pm 2.24}$

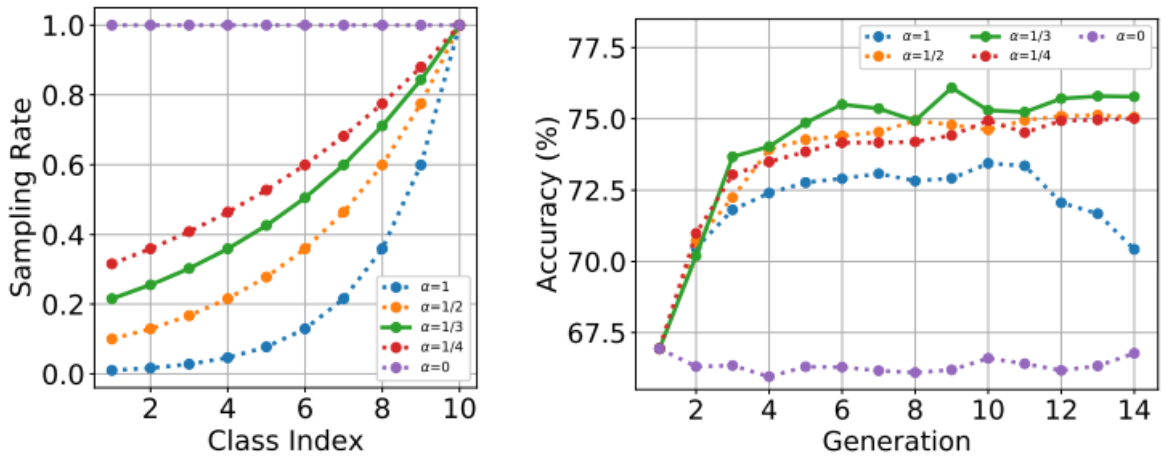
Table 2. We compare CReST and CReST+ with baseline methods including different SSL algorithms and typical class-rebalancing techniques designed for fully-supervised learning. For fair comparison, all models are measured at the same number of training steps. See text for details. Three imbalance ratios γ with $\beta = 10\%$ labels are evaluated. Numbers are averaged over 5 different folds.

Method	Gen_1	Gen_2	Gen_3
Supervised (100% labels)	75.8	-	-
Supervised (10% labels)	46.0	-	-
FixMatch (10% labels)	65.8	-	-
w/ DA ($t = 0.5$)	69.1	-	-
w/ CReST	65.8	67.6	67.7
w/ CReST+	68.3	70.7	**73.7**

Table 4. Evaluating the proposed method on ImageNet127 with $\beta = 10\%$ samples are labeled. We retrain FixMatch models for 3 generations with our CReST and CReST+.

Experiment-sampling rate hyper-parameter α

$$\mu_l = \left(\frac{N_{L+1-l}}{N_1}\right)^{\alpha}$$



(a) (b)

Figure 4. Effect of α across multiple generations on CIFAR10-LT ($\gamma = 100$, $\beta = 10\%$) in CReST. (a) Illustration of how α influences sampling rate. (b) Test accuracy over generations with different α. When $\alpha = 0$, the method falls back to conventional self-training with all the unlabeled examples and corresponding pseudo-labels added into the labeled set, showing no improvement after generations of retraining, whereas our class-rebalancing sampling ($\alpha > 0$) helps.

imbalance ratio γ = 100
label fraction β = 10%

Experiment-temperature t

imbalance ratio γ = 100
label fraction β = 10%

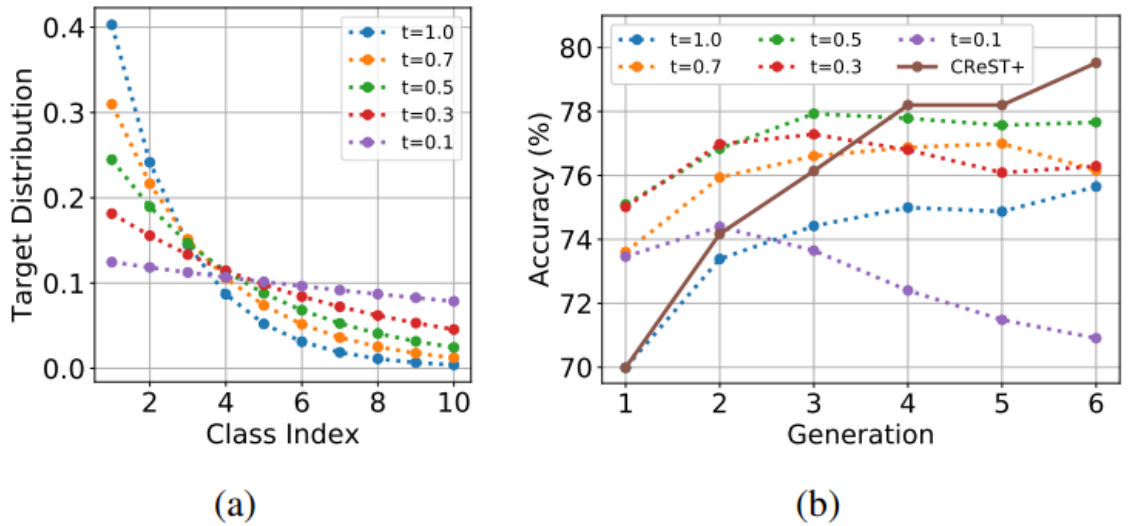


Figure 5. Effect of temperature t across multiple generations on CIFAR10-LT ($\gamma = 100$, $\beta = 10\%$). (a) Illustration of how t controls the target distribution of distribution alignment. (b) Test accuracy over generations with different constant t and our CReST+ using progressive t. Compared to using a constant t, CReST+ achieves the best final accuracy by progressing from $t = 0$ to $t_{\text{min}} = 0.5$ over 6 generations.

Expreiment-per class recall

Method / Class	Split	1	2	3	4	5	6	7	8	9	10	Avg.
FixMatch [39]	test	**98.7**	**99.5**	**90.0**	**83.5**	85.0	47.6	69.9	59.0	8.9	7.2	64.9
w/ CReST	test	97.7	98.3	88.8	81.9	**88.2**	59.7	79.5	61.2	47.0	47.9	75.0
		-1.0	-1.2	-1.2	-1.6	+3.2	+12.1	+9.6	+2.2	+38.1	+40.7	+10.1
w/ CReST+	test	93.8	97.7	87.3	76.9	87.5	**69.2**	**84.9**	**67.9**	**60.3**	**70.8**	**79.6**
		-4.9	-1.8	-2.7	-6.6	+2.5	+21.6	+15.0	+8.9	+51.4	+63.6	+14.7
FixMatch [39]	unlabeled	**98.5**	**99.1**	**90.0**	**84.0**	84.7	49.7	64.9	65.6	14.9	22.2	67.4
w/ CReST	unlabeled	97.8	96.8	**90.0**	82.9	87.4	62.4	79.3	64.8	60.8	66.7	78.9
		-0.7	-2.3	0	-1.1	+2.7	+12.7	+14.4	-0.8	+45.9	+44.5	+11.5
w/ CReST+	unlabeled	92.2	95.7	86.1	76.7	**87.6**	**68.1**	**85.1**	**71.2**	**75.7**	**75.6**	**81.4**
		-6.3	-3.4	-3.9	-7.3	+2.9	+18.4	+20.2	+5.6	+60.8	+53.4	+14.0

Table 5. Per-class recall (%) on the balanced test set and the imbalanced unlabeled set of CIFAR10-LT ($\gamma = 100$, $\beta = 10\%$). Our strategies compromise small loss on majority classes for significant gain on minority classes, leading to improved averaged recall over all classes.

The End