1

Distribution-Balanced Loss for Multi-Label Classification in Long-Tailed Datasets

ECCV 2020 spotlight

pdf

Motivation

- *Problems in multi-label recognition problems*
  - Long-tailed distribution of different object categories typically in practical contexts.
  - Individual images can generally be associated with multiple semantic labels.

- *A widely adopted solution*
  - Use ***binary cross-entropy*** to replace the softmax loss.
  - Use ***class-specific re-weighting*** to balance the contributions of different classes, *e.g.,* setting the class weights to be inversely proportional to the class size.

Motivation

- *Problems in the widely adopted solution*

  - Label co-occurrence.

    - Example: “tigers” and “leopards”, is likely to be associated with more common labels like “trees” and “river”.

  - Dominance of ***negative*** labels.

    - Positive and negative classes are ***treated uniformly*** in symmetric Binary Cross Entropy (BCE) loss.
    - ***Over-suppression*** of the negative side resulted from the dominant portion of negative classes, introducing significant bias to the classification boundaries.

Distribution-Balanced Loss

- **Re-balanced weighting**
  - Adjust the weights in a way that closes the gap between expected sampling times and actual sampling times, with label co-occurrence taken into account.
- ***Negative-tolerant regularization***
  - Avoid over-suppression of the negative labels by setting a margin and a re-scaling factor.

Distribution-Balanced Loss

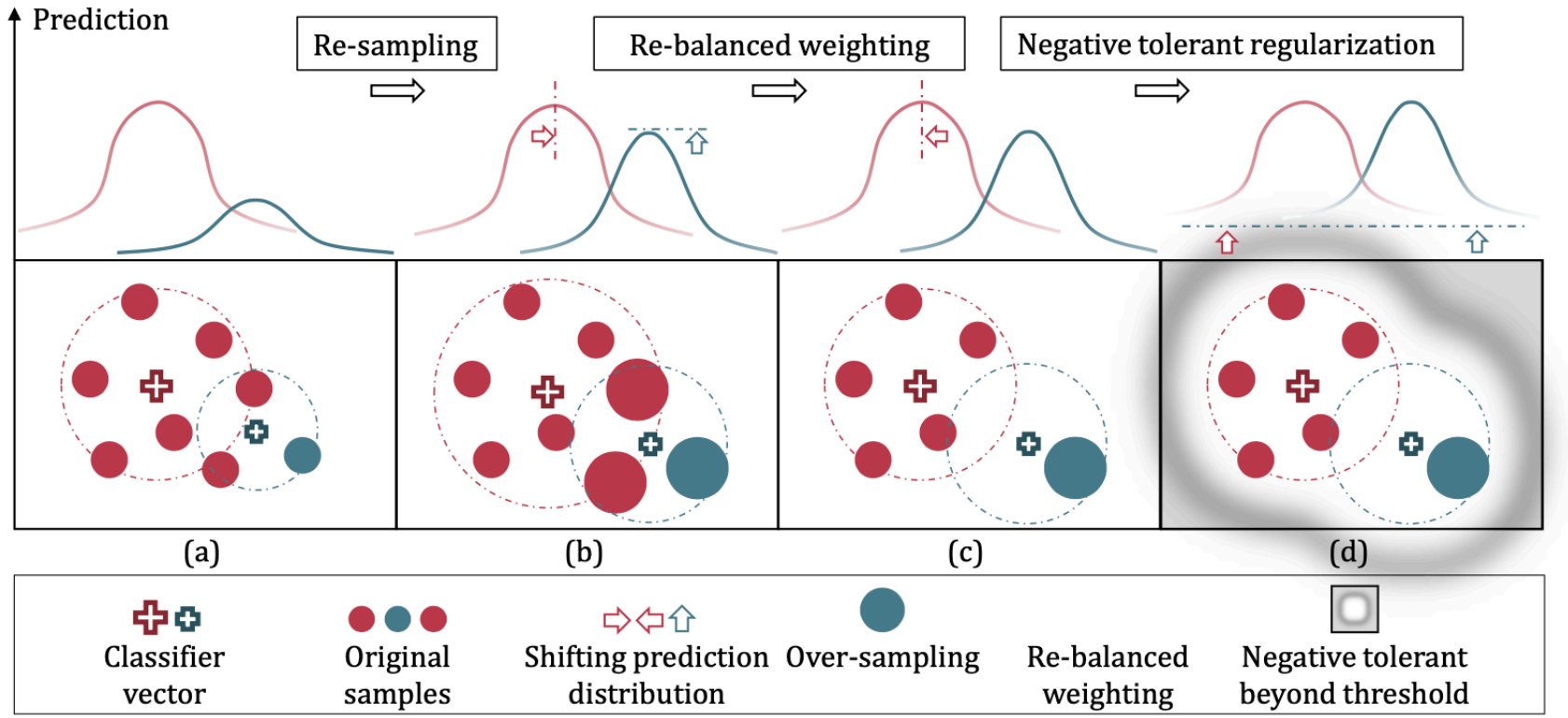


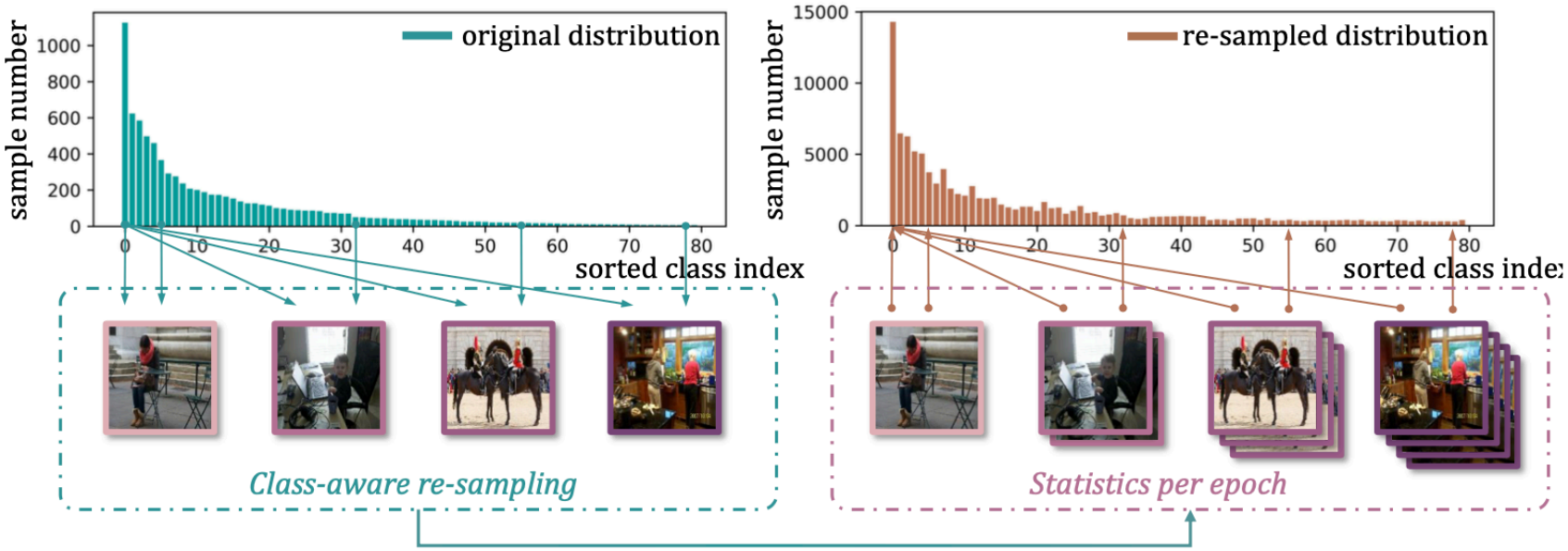
Fig. 1: Our *Distribution-Balanced Loss* performs re-balanced weighting along with re-sampling that takes label co-ocurrence into consideration, and it leverages negative-tolerant regularization to avoid over-suppression of the negative labels caused by the dominance of negative classes in *binary cross entropy (BCE)*

Distribution-Balanced Loss

- *Preliminaries*
  - The dataset $\mathcal{D} = \{(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^N, \mathbf{y}^N)\}$ has N training samples.
  - $\mathbf{y}^k = [y_1^k, \ldots, y_C^k] \in \{0, 1\}^C$, C is the number of classes.
  - $n^i = \sum_{k=1}^{N} y_i^k$ denotes the number of training examples that contain class i.

Distribution-Balanced Loss

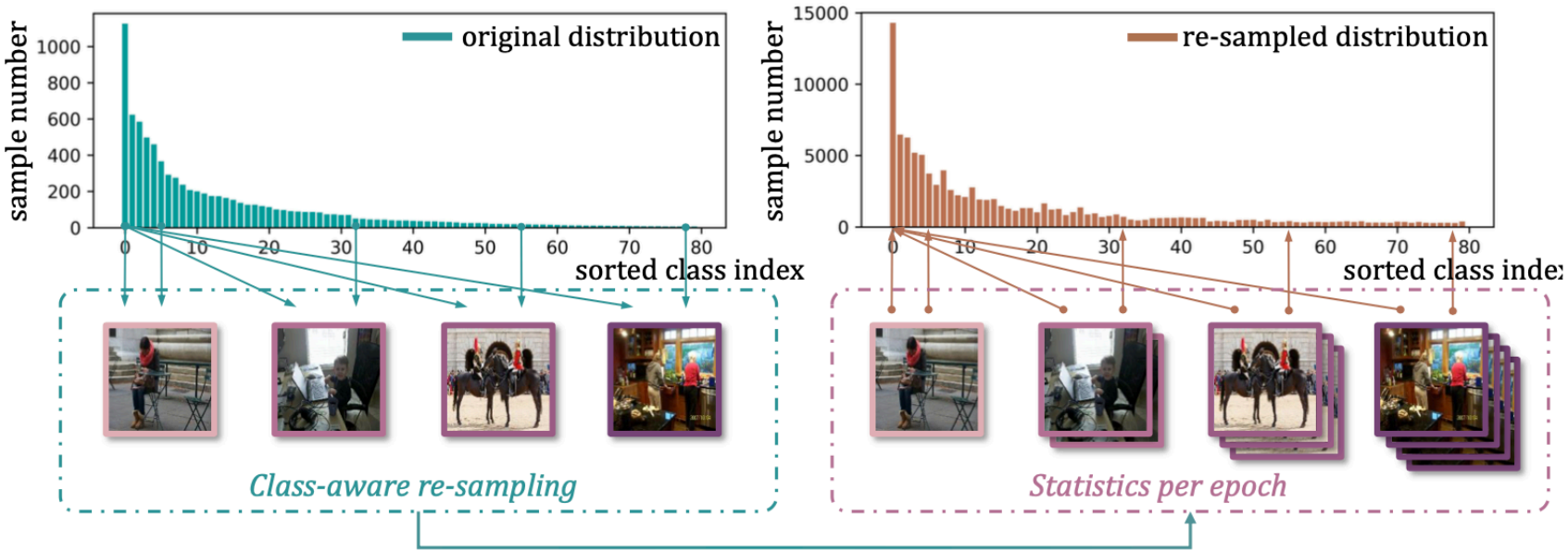
- *Re-balanced weighting after re-sampling*
  - Common sampling rule: select each example from training set with equal probability.
  - Re-sampling instances from one specific class will inevitably influence the sample numbers of the other classes co-occurring.



Distribution-Balanced Loss

- *Re-balanced weighting after re-sampling*
  - $p(i|j)$ is the conditional probability of an instance containing label i under the condition of containing label j.

$$\hat{p}_i = \frac{1}{C}\sum_{j=0}^{C} p(i|j) = \frac{1}{C}\sum_{j=0}^{C} \frac{n_{i\cap j}}{n_j}$$



Distribution-Balanced Loss

- *Re-balanced weighting after re-sampling*
  - First, without taking label co-occurrence into consideration, for each instance k and class i with $y_i^k = 1$, the expectation of class-level sampling frequency can be calculated as $P_i^C(x^k)$.
  - Then, given an instance x^k and its corresponding label y^k , it is supposed to be repeatedly sampled by each positive class i it contains, thus the expectation of instance-level sampling frequency can be estimated as $P_i^I(x^k)$.

$$P_i^C(x^k) = \frac{1}{C}\frac{1}{n_i},\ P^I(x^k) = \frac{1}{C}\sum_{y_i^k=1}\frac{1}{n_i}$$

Distribution-Balanced Loss

- *Re-balanced weighting after re-sampling*
  - Define a re-balancing weight, namely $r_{\mathrm{i}}^{\mathrm{k}}$, to close the gap between expected sampling times and actual sampling times.

$$r_i^k = \frac{P_i^C(x^k)}{P^I(x^k)}$$

  - Further design a smoothing function to map r into a proper range of values.

$$\hat{r} = \alpha + \frac{1}{1 + exp(-\beta \times (r - \mu))}$$

Here α is an overall lift in weight, while β and μ controls the shape of the mapping function, which rapidly increases near 0 and goes flat near 1.

Distribution-Balanced Loss

- *Re-balanced weighting after re-sampling*
  - The loss function, Re-balanced-BCE, where z_k denotes the output of the classifier,

$$\mathcal{L}_{R-BCE}(x^k, y^k) = \frac{1}{C}\sum_{i=0}^{C}\left[y_i^k log(1+e^{-z_i^k}) + (1-y_i^k) log(1+e^{z_i^k})\right] \times \hat{r}_i^k$$

Distribution-Balanced Loss

- *Negative-tolerant regularization*

o Defects of BCE loss

- BCE loss: Considers the recognition task as a series of binary classification tasks, calculating independent class-wise probability with sigmoid function，sometimes suffers from over-suppression for negative labels.
- Sigmoid: Treats positive and negative classes independently and encourages the logits of both positive and negative classes to be away from zero in the same gradient declining manner.

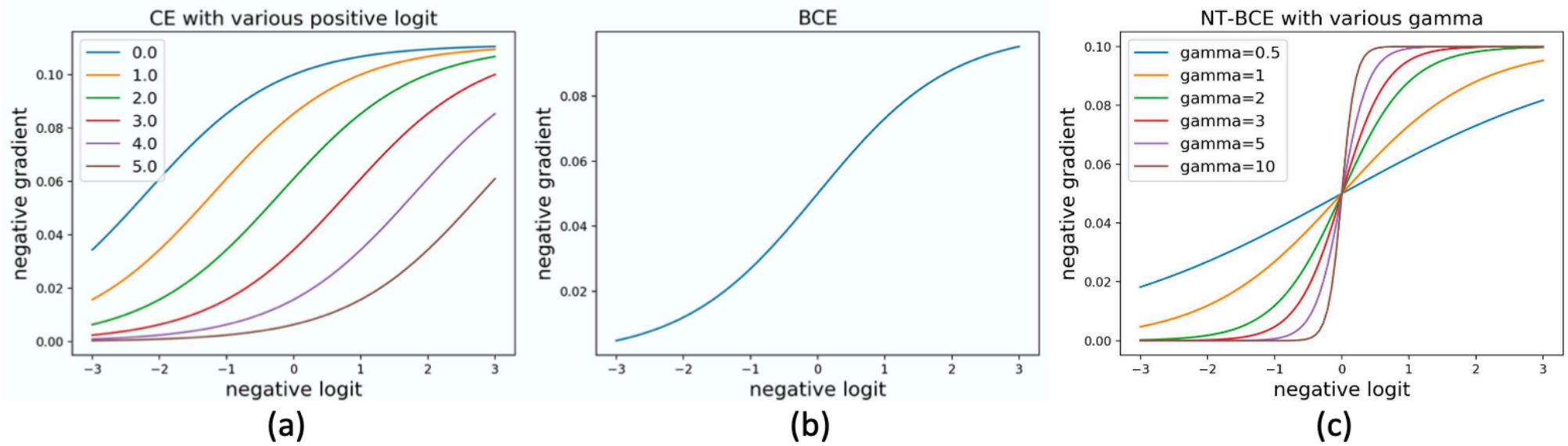
o Defects of CE loss

- Cross Entropy (CE) loss: Utilizes softmax to emphasize mutual exclusion and is popular in single-label classification.
- Softmax: The optimization step would be rather small once the logit for positive class is much higher than those of negative classes.

Distribution-Balanced Loss

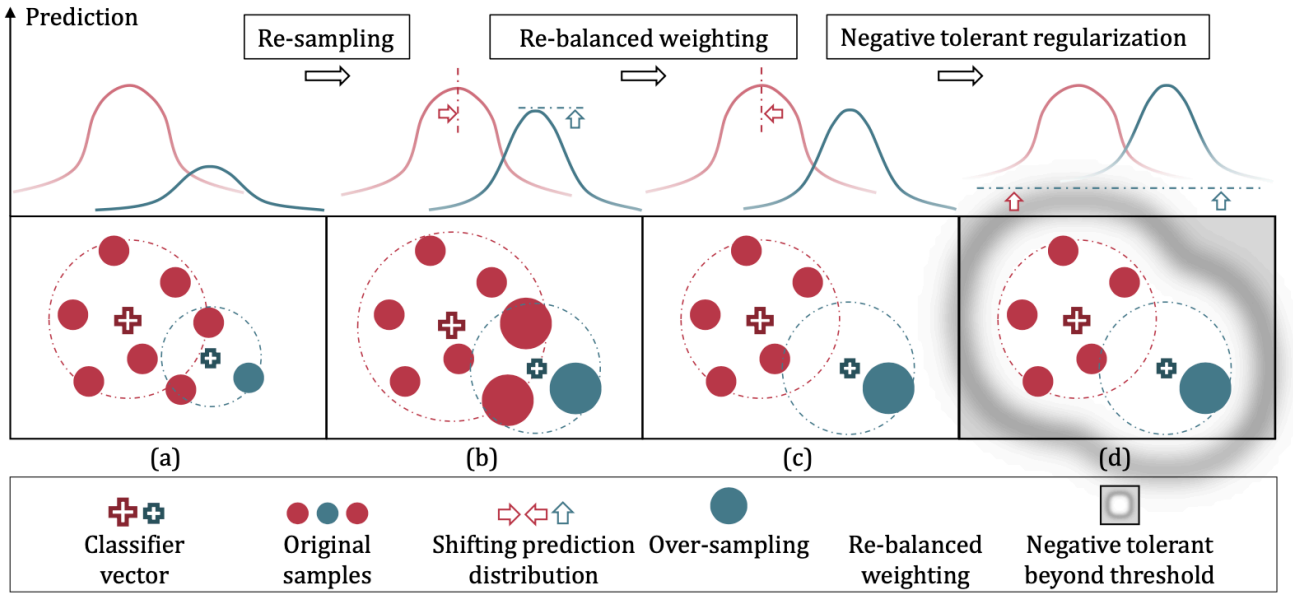
- *Negative-tolerant regularization*
  - Defects of BCE and CE

$$\begin{cases} \dfrac{\partial \mathcal{L}_{CE}(z_j, y)}{\partial (z_j)} = \dfrac{e^{z_j}}{\sum_{i=0}^{C} e^{z_i}}, y_j = 0 \\ \dfrac{\partial \mathcal{L}_{BCE}(z_j, y)}{\partial (z_j)} = \dfrac{1}{C} \dfrac{e^{z_j}}{1 + e^{z_j}}, y_j = 0 \end{cases}$$



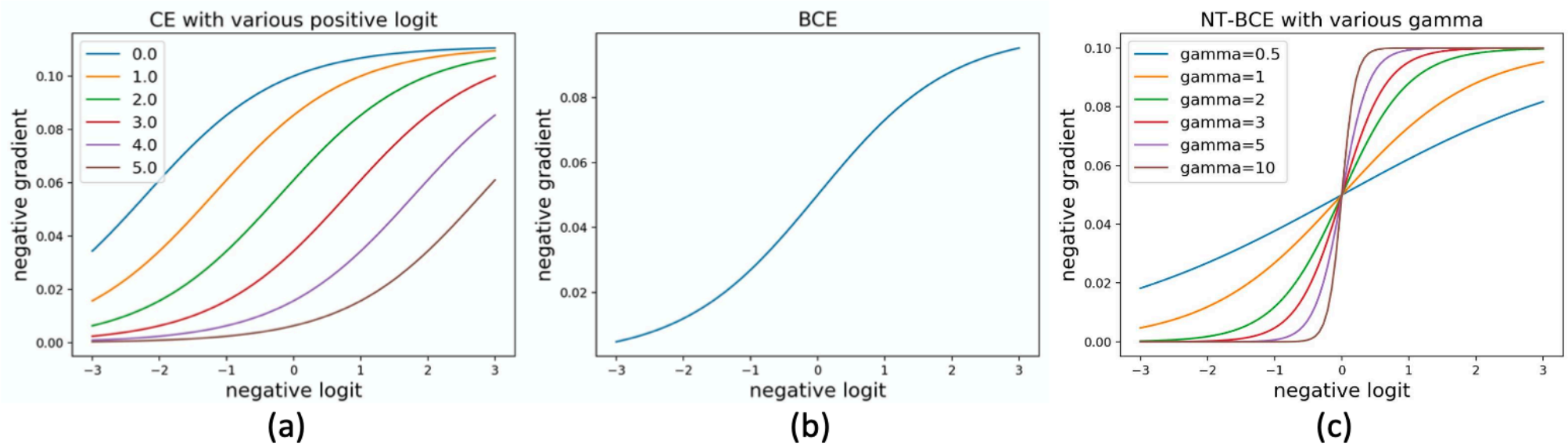
Distribution-Balanced Loss

- *Negative-tolerant regularization*
  - Consequences in long-tailed classification
    - The classifiers for the tail classes would over-fit to a limited number of positive samples in the feature space.
    - The classifiers push a huge number of negative samples away to produce lower logits.



Distribution-Balanced Loss

- *Negative-tolerant regularization*
  - A regularization to overcome the over-suppression.
  - Idea: The loss by negative logits actually needs a sharp drop once it is optimized to be lower than a threshold so that they will not be continuously suppressed due to a relatively small gradient.



Distribution-Balanced Loss

- *Negative-tolerant regularization*
  - A regularization to overcome the over-suppression.
    - How to eliminate over-suppression:

1. Use a non-zero bias initialization to act as the thresholds.
2. Then apply a linear scaling to the negative logits before their calculation in the standard BCE, together with a regularization parameter to constrain the gradient between 0 and 1.

$$\mathcal{L}_{NT-BCE}(x^k, y^k) = \frac{1}{C}\sum_{i=0}^{C} y_i^k log(1 + e^{z_i^k - \nu_i}) + \frac{1}{\lambda}(1 - y_i^k) log(1 + e^{-\lambda(z_i^k - \nu_i)})$$

λ is the scale factor that effects the loss gradient, ν is a class-specific bias.

Distribution-Balanced Loss

- *Negative-tolerant regularization*
  - A regularization to overcome the over-suppression.
  - How to estimate the bias ν:

At the very beginning of training, considering the bias b_i as the only variable, and assuming the class prior to be $p_i = n_i/N_0$, an approximation of averaged loss by class i can be deduced,

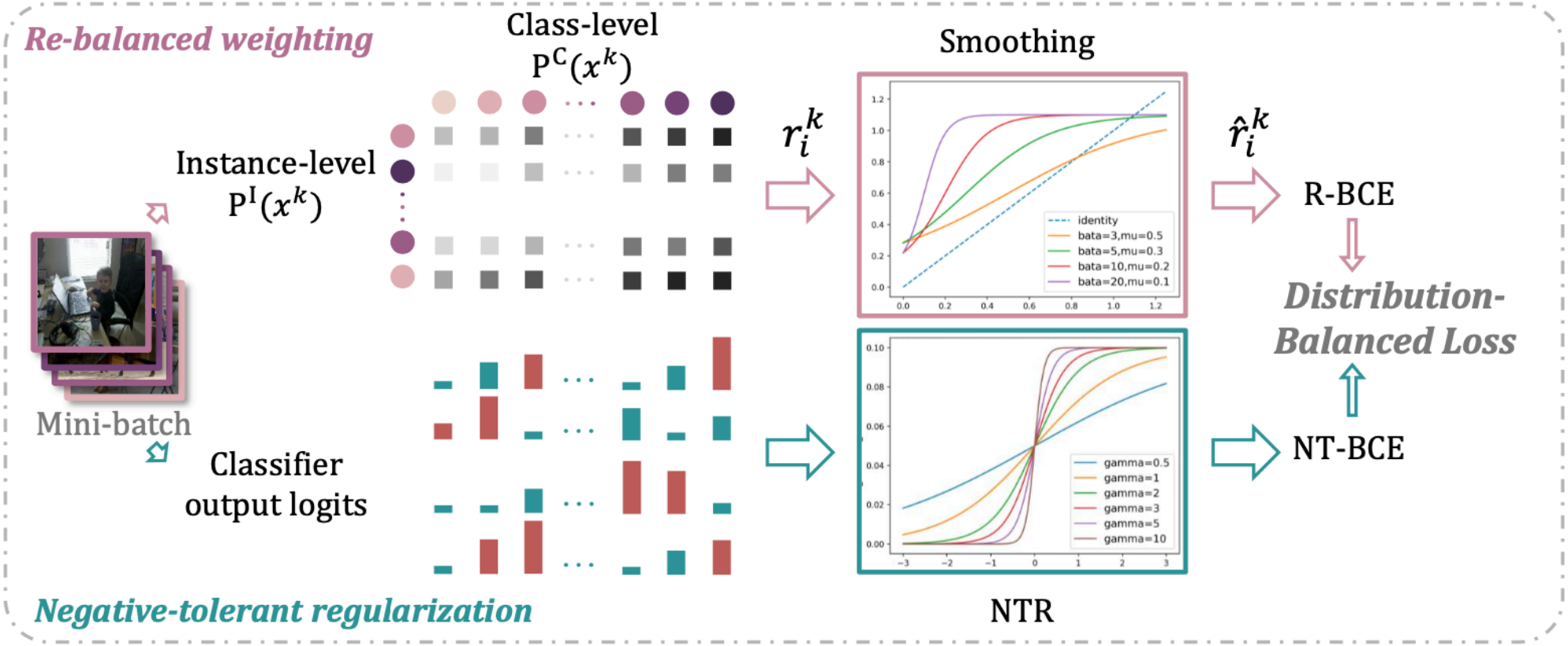
$$L_i = p_i \log(1 + e^{-b_i}) + (1 - p_i) \log(1 + e^{b_i})$$

$$\hat{b}_i = -\log(\frac{1}{p_i} - 1),\ \nu_i = -\kappa\ \hat{b}_i$$

Distribution-Balanced Loss

- *Integrated loss function*

$$\mathcal{L}_{DB}(x^k, y^k) = \frac{1}{C}\sum_{i=0}^{C} \hat{r}_i^k \left[y_i^k log(1 + e^{z_i^k - \nu_i}) + \frac{1}{\lambda}(1 - y_i^k) log(1 + e^{-\lambda(z_i^k - \nu_i)}) \right]$$



Experiment

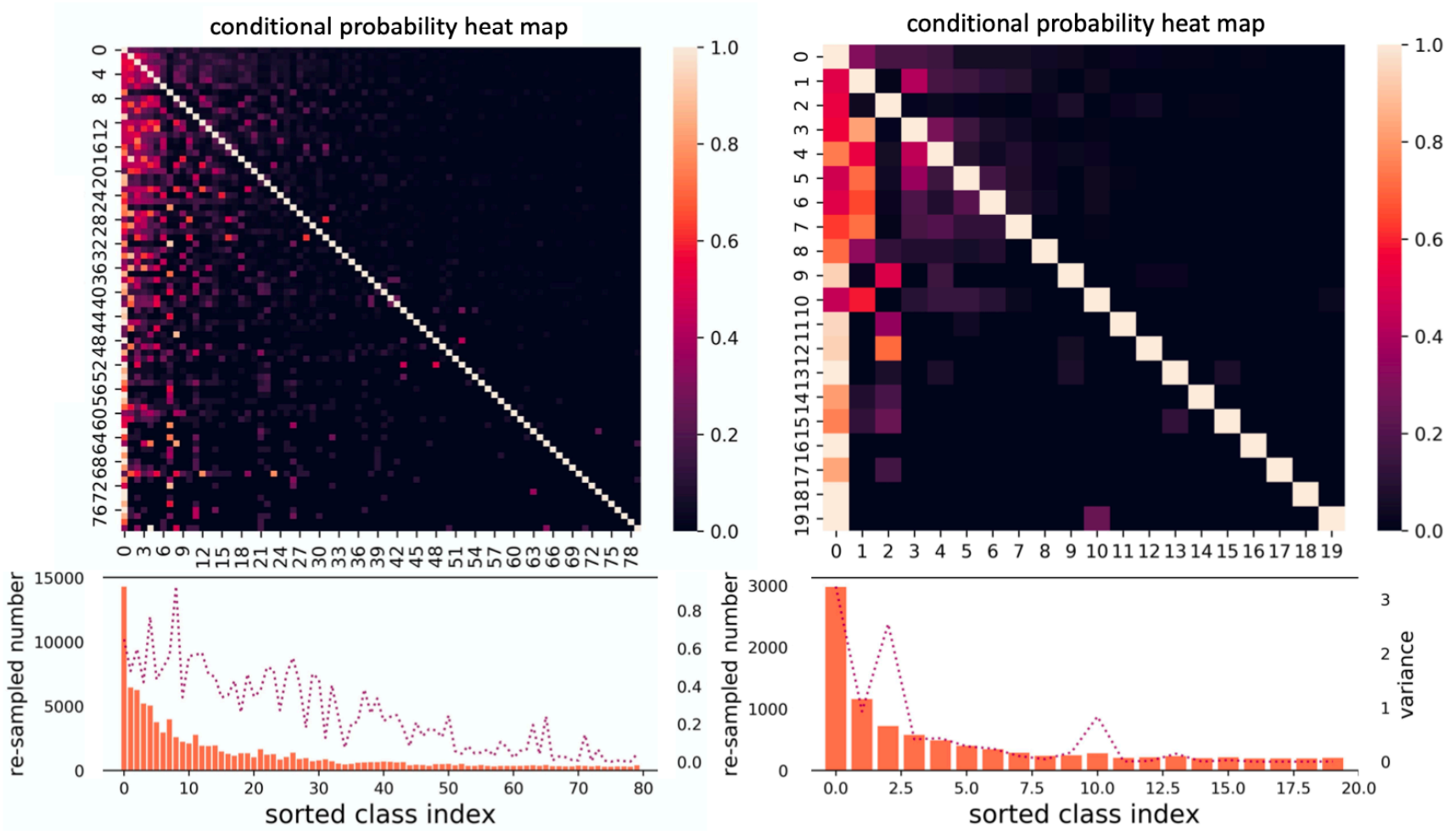
- *Results on datasets VOC-MLT and COCO-MLT*

Table 1: Experimental results of mAP by our methods and other comparing approaches on VOC-MLT and COCO-MLT. We evaluate the results on the whole class set and the three subsets, respectively

Datasets	VOC-MLT				COCO-MLT			
Methodss	total	head	medium	tail	total	head	medium	tail
ERM	70.86	68.91	80.20	65.31	41.27	48.48	49.06	24.25
RW	74.70	67.58	82.81	73.96	42.27	48.62	45.80	32.02
Focal Loss [21]	73.88	69.41	81.43	71.56	49.46	49.80	54.77	42.14
RS [30]	75.38	70.95	82.94	73.05	46.97	47.58	50.55	41.7
RS-Focal	76.45	72.05	83.42	74.52	51.14	48.90	54.79	48.30
ML-GCN [4]	68.92	70.14	76.41	62.39	44.24	44.04	48.36	38.96
LDAM [3]	70.73	68.73	80.38	69.09	40.53	48.77	48.38	22.92
CB-Focal [5]	75.24	70.30	83.53	72.74	49.06	47.91	53.01	44.85
R-BCE	76.34	71.40	82.76	75.22	49.43	48.77	53.00	45.33
R-BCE-Focal	77.39	72.44	83.16	76.77	52.75	50.20	56.52	50.02
DB	78.65	73.16	84.11	78.66	52.53	50.25	56.33	49.54
DB-Focal	**78.94**	**73.22**	**84.18**	**79.30**	**53.55**	**51.13**	**57.05**	**51.06**

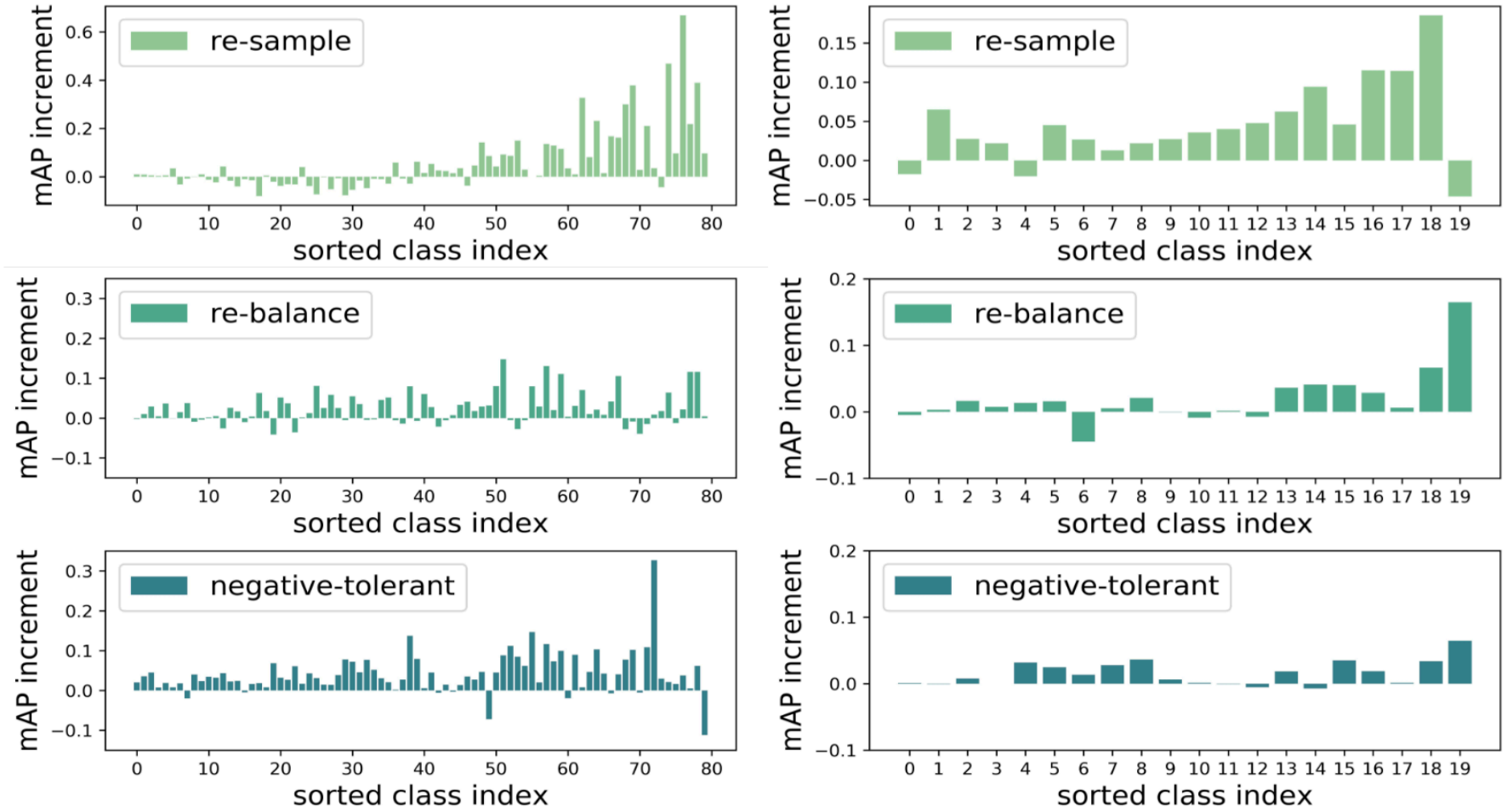
Experiment

- *Visualization of the Imbalance caused by Re-sampling*



Experiment

- *Step-wise Evaluation*



Experiment

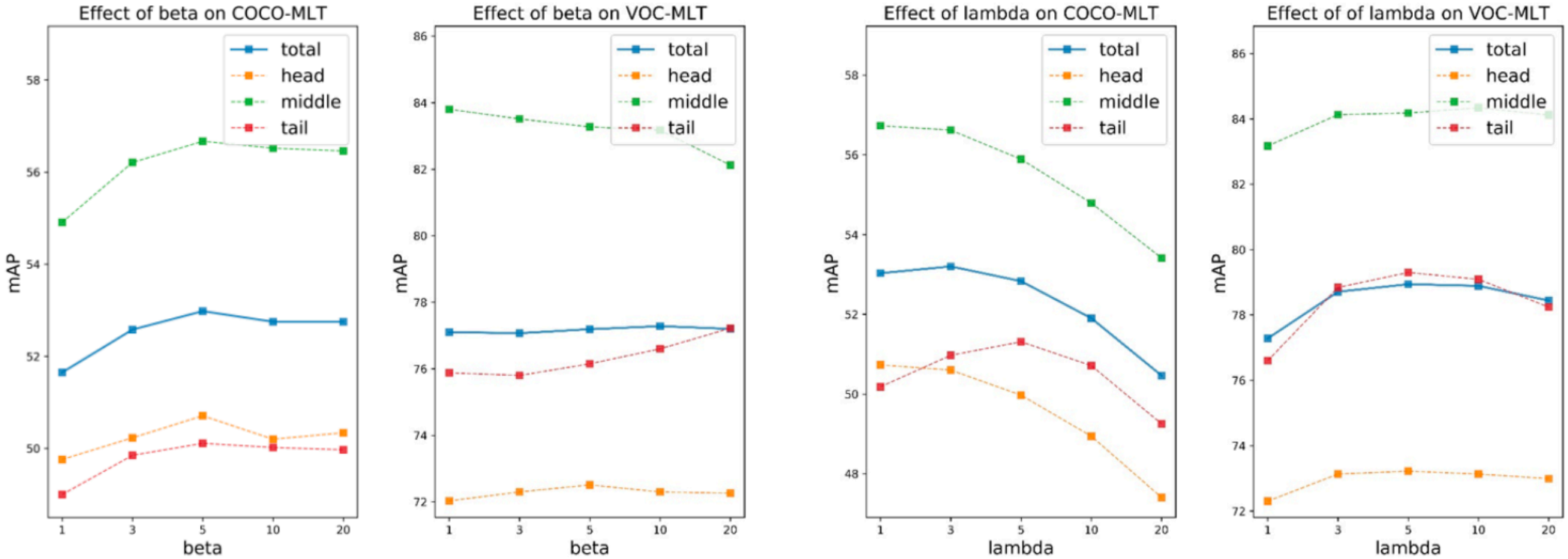
- *The Combination of Re-sampling and Various Re-weighting Methods*

Table 2: Experimental results on re-sampling combined with several re-weighting techniques. CB loss with focal is reported by [5] to perform better, so all the other techniques are enhanced with focal loss for fair comparison

Datasets	VOC-MLT				COCO-MLT			
Methodss	total	head	medium	tail	total	head	medium	tail
RS [30]	75.38	70.95	82.94	73.05	47.7	46.44	51.93	43.23
RS-Focal [21]	76.45	72.05	83.42	74.52	51.14	48.9	54.79	48.30
RS+RW-Focal [28,27]	71.96	63.14	81.09	71.73	49.07	47.80	52.09	46.20
RS+CB-Focal [5]	75.24	70.30	**83.53**	72.74	50.07	48.45	54.03	48.28
R-BCE-Focal	**77.39**	**72.44**	83.16	**76.77**	**52.75**	**50.2**	**56.52**	**50.02**

Experiment

- *Effects of parameters in DB loss*



2

Visual Relation Grounding in Videos

ECCV 2020 spotlight

pdf

Task

- *Visual Relation Grounding in Videos (vRGV)*
  - The task takes a relation in the form of subject-predicate-object as query, and requires the models to localize the related visual subject and object in a video by returning their trajectories.

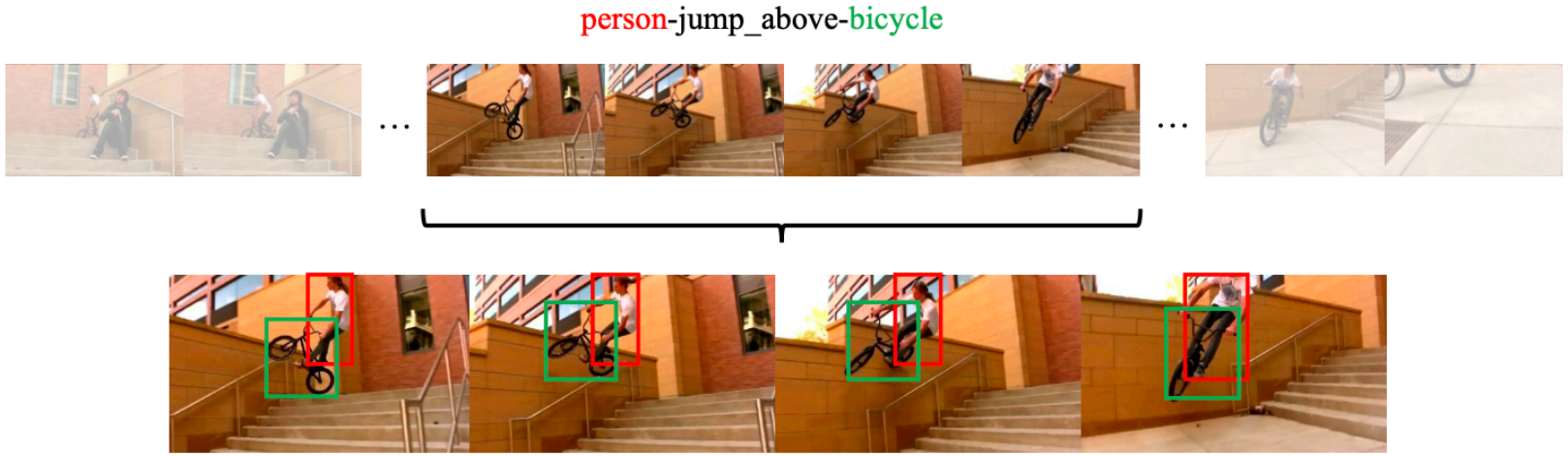
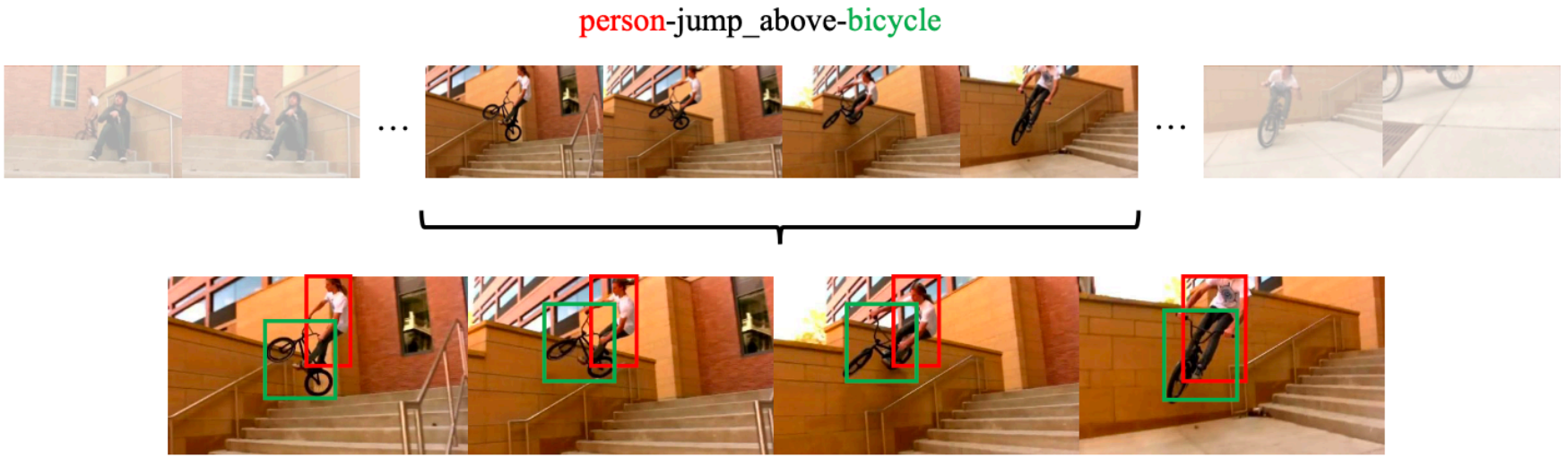


Fig. 1: Illustration of the vRGV task. For the query relation *person-jump_above-bicycle* and an untrimmed video containing the relation, the objective is to find a video segment along with two trajectories corresponding to the subject (red box) and object (green box) that match the query relation.

Task

- *Challenges in vRGV*
  - Both the subject and object are required to be ***spatio-temporally localized*** to ground a query relation.
  - The temporal ***dynamic*** nature of visual relations is difficult to capture.
  - The grounding should be achieved under ***no direct supervision*** in space and time.



Method

- *Task definition*
  - Given a set of query relations in form of $R = \{< S - P - O >\}$ and a set of untrimmed videos V, where S, P, O denote the subject, predicate and object, respectively.
  - Each specific query relation R_i is coupled with several videos from V which contain that relation.
  - The task is to spatio-temporally localize in the videos the respective subjects and objects by returning their trajectories T_s, T_o.
  - The trajectory T is given by a sequence of bounding boxes tied to a certain visual entity across a video segment.

Method

- *Solution Overview*

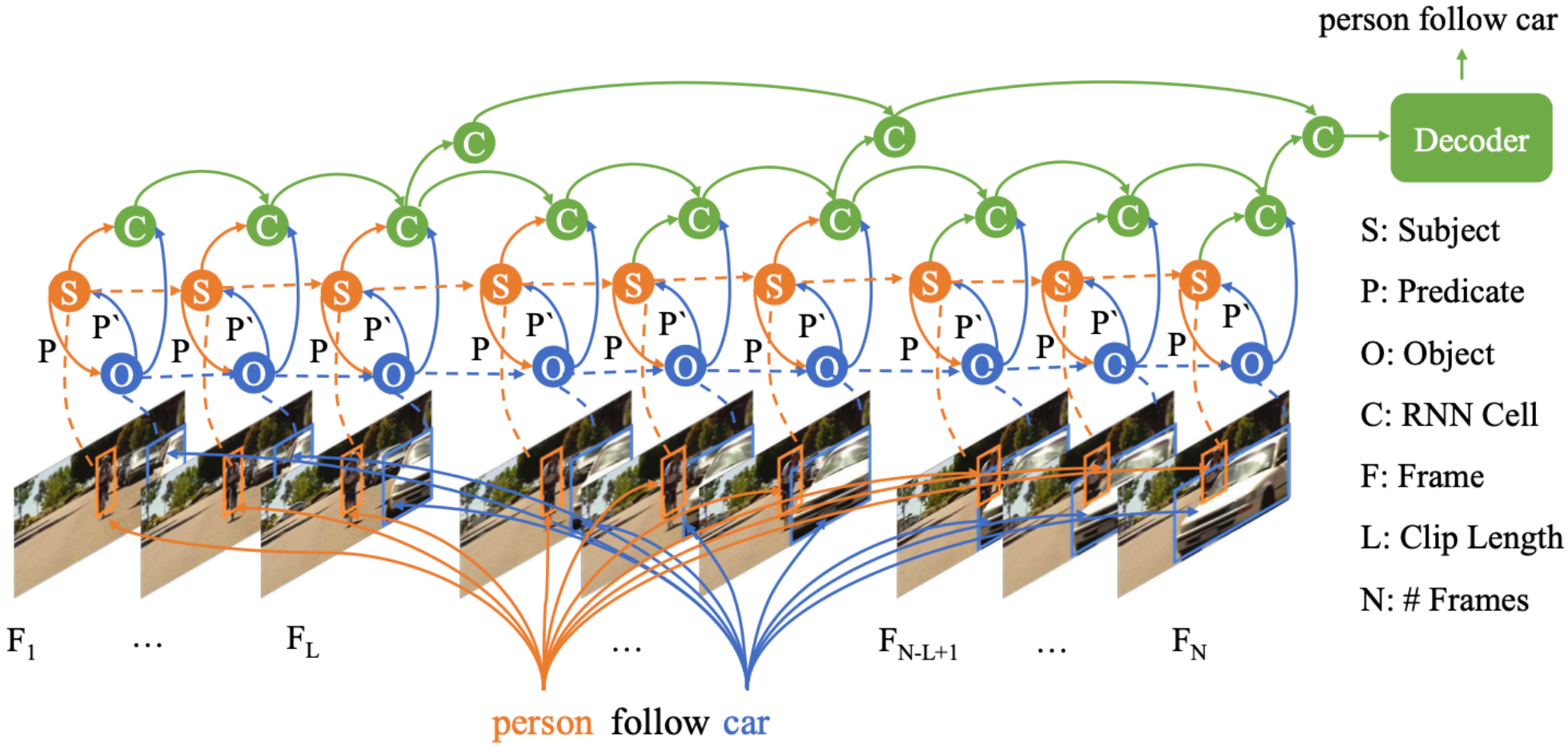
  - Given a video of N frames and extracted M region proposals for each frame. A video can be represented by a set of regions $V = \{B_{i,j} \mid i \in [1, N],\ j \in [1, M]\}$, and a trajectory $T = \{B_i \mid i \in [k, l], k \in [1, N], l \in [k, N]\}$ can be a sequence of bounding boxes in the video.

  - Goal: Learn to ground a given relation R by finding two trajectories T_s, $T_\mathbf{o}$ that indicate the subject and object of the relation by by maximizing the following posterior probability,

$$T_s^*, T_o^* = \underset{T_s,\ T_o}{\arg\max}\, P(R \mid T_s,\ T_o) * P(T_s,\ T_o \mid V,\ R)$$

Method

- *Solution Overview*



Method

- *Message Passing by Attention Shifting*

  - Spatial attention

    - Input: All the region proposals $B = \{B_j \mid j \in [1, M]\}$ in a frame and the query relation $R = \{< S - P - O >\}$.
    - Spatial attention unit (SAU) learns two spatial attentions ($\alpha_s^{M\times1}$, $\alpha_o^{M\times1}$) corresponding to the subject and object,

$$\alpha_s = SAU(\ f(B),\ g(S)\ ),\ \alpha_o = SAU(\ f(B),\ g(O)\ )$$

      where $g(\cdot)$ returns the textual word feature for subject (S) or object (O),

$f(\cdot)$ means feature extraction for the region proposals.

Method

- *Message Passing by Attention Shifting*
  - Spatial attention
    - $f(\cdot)$ is related to the object appearances, consisting of transformed ROI-aligned feature from object detection models, *i.e.*, $f_{app} = CNN(B_j)$, relative locations and sizes $f_B = \left[\frac{x_{min}}{W}, \frac{y_{min}}{W}, \frac{x_{max}}{W}, \frac{y_{max}}{W}, \frac{area}{W*H}\right]$, $f(B_j) = f_{app} + f_B$.
    - Given the textual subject representation $g(S)$ and each region proposal $f(B_j)$, the attention score s_j is obtained,

$$s_j = W_2 tanh(W_1[f(B_j),\ g(S)] + b_1)$$

$$\alpha_{s_j} = softmax(s_j) = \frac{exp(s_j)}{\sum_{z=1}^{M} exp(s_z)},\ f_s = \sum_{j=0}^{M} \alpha_{s_j} f(B_j)$$

Method

- *Message Passing by Attention Shifting*

  - Attention shifting

  - Two independent transfer matrices W_{so} and W_{os} are tied to the forward relationship (P, message from subject to object) and backward relationship (P', message from object to subject), respectively.

$$f_{so} = ReLU(W_{so}\alpha_s),\ f_{os} = ReLU(W_{os}\alpha_o)$$
$$f_s = f_s + f_{os},\ f_o = f_o + f_{so}$$

  - Finally, the subject and object representations will be concatenated and transformed to obtain the node input at time step i, *i.e.*, $f_i = W_3([f_s, f_o]) + b_3$, where W_3, b_3 are learnable parameters.

Method

- *Hierarchical Temporal Attention*
  - A video is divided into $H = \frac{N}{L}$ short clips of length L. Two relation-aware hierarchical temporal attention units TAU_1 andTAU_2.
  - The frame-wise temporal attention β^{l1}(of dimension N) is obtained with the sequence of frame-wise feature f^{l1} and the output at the last time step of the clip-level neural encoder f_H^{l2},

$$\beta^{l1} = TAU_1(f^{l1},\ f_H^{l2}),\ f_i^{l1} = LSTM_{l1}(f_{1,\cdots,i}),\ i \in [1, N]$$

  - The sequence of clip-level inputs f_c are obtained by selecting the output of the first layer of LSTM at every L steps, *i.e.*, $f^c = \{f_i^{l1} | \ i \in \{1, L, \cdots, N\}\}$. β^{l2} (of dimension H) is the clip-level temporal attention distribution,

$$\beta^{l2} = TAU_2(f^c,\ f_R),\ f_H^{l2} = LSTM_{l2}((\beta_i^{l2} f_i^c)_{i=1,\cdots,H})$$

Method

- *Training*
  - The final graph embedding by an attention-guided pooling of the node representations across the video, *i.e.*, $feat_v = \sum \beta^{l1} f^{l1}$, is derived.
  - The model was trained with the cross-entropy loss,

$$L_{rec} = -\frac{1}{n_{vr}} \sum_{n=1}^{n_{vr}} \sum_{t=1}^{n_w} logP(R_t | R_{0:t-1},\ feat_v)$$

where R_t denotes the t^{th} word in the relation, n^{vr} and n^w denote the number of video-relation samples and number of words in the relation, respectively.

Method

- *Inference*
  - Set a temporal threshold to obtain a set of candidate sub-segments for each relation-video instance in the test set.
  - Then, a linking score $s(B_{i,p}, B_{i+1,q})$ between regions of successive frames is defined (after sampling),

$$s(B_{i,p}, B_{i+1,q}) = \alpha_{i,p} + \alpha_{i+1,q} + \lambda \cdot IoU(B_{i,p}, B_{i+1,q})$$

    where α is the spatial attention value.
  - The final trajectory can thus be achieved by finding the optimal path over the segment

$$T^* = \arg\max_T \frac{1}{K-1} \sum_{i=1}^{K-1} s(B_{i,p}, B_{i+1,q})$$

Experiment

- *Results on ImageNet-VidVRD dataset*

Table 2: Results of visual relation grounding in videos. We add bold and underline to highlight the best and second-best results under each metric respectively.

Methods	sIoU=0.3			sIoU=0.5			sIoU=0.7			Average		
	Acc_S	Acc_O	Acc_R	Acc_S	Acc_O	Acc_R	Acc_S	Acc_O	Acc_R	Acc_S	Acc_O	Acc_R
T-Rank V_1 [3]	33.55	27.52	17.25	22.61	12.79	4.49	6.31	3.30	0.76	20.27	10.68	3.99
T-Rank V_2 [3]	34.35	21.71	15.06	23.00	9.18	3.82	7.06	2.09	0.50	20.83	7.35	3.16
Co-occur* [15]	27.84	25.62	18.44	23.50	20.40	13.81	17.02	14.93	7.29	22.99	19.33	12.80
Co-occur [15]	31.31	30.65	21.79	28.02	27.69	18.86	<u>21.99</u>	<u>21.64</u>	<u>13.16</u>	25.90	25.23	16.48
vRGV* (ours)	<u>37.61</u>	<u>37.75</u>	<u>27.54</u>	<u>32.17</u>	<u>32.32</u>	<u>21.43</u>	21.34	21.02	10.62	<u>31.64</u>	<u>30.92</u>	<u>20.54</u>
vRGV (ours)	**42.31**	**41.31**	**29.95**	**37.11**	**37.52**	**24.77**	**29.71**	**29.72**	**17.09**	**36.77**	**36.30**	**24.58**

Experiment

- *Qualitative results*

Fig. 3: Qualitative results on the query relation *bicycle-move_beneath-person*.

Experiment

- *Qualitative results*

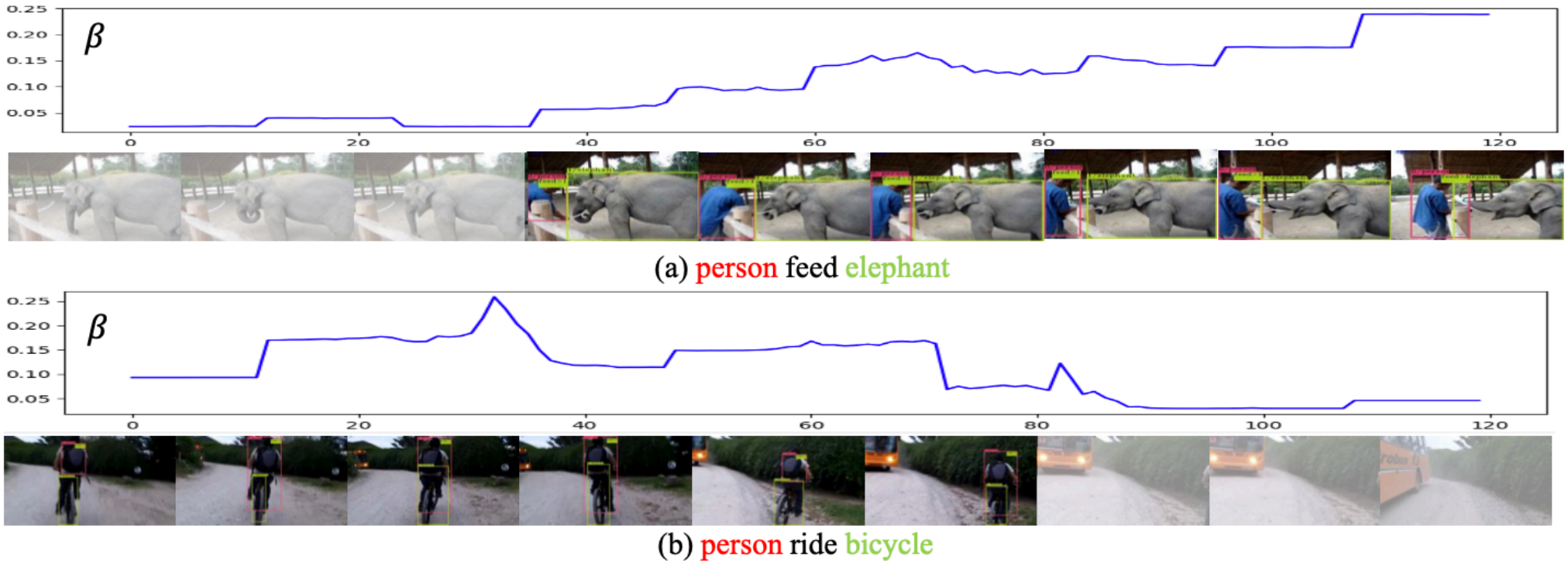


Fig. 4: Qualitative results based on temporal threshold 0.04.

Experiment

- *Ablation Studies*

Table 3: Model ablation results on ImageNet-VidVRD.

Models	sIoU=0.3			sIoU=0.5			sIoU=0.7			Average		
	Acc_S	Acc_O	Acc_R	Acc_S	Acc_O	Acc_R	Acc_S	Acc_O	Acc_R	Acc_S	Acc_O	Acc_R
vRGV	**42.31**	**41.31**	**29.95**	**37.11**	**37.52**	**24.77**	**29.71**	**29.72**	**17.09**	**36.77**	**36.30**	**24.58**
w/o Msg	34.72	33.23	23.96	31.60	29.15	19.43	22.56	21.36	11.78	29.41	27.46	17.63
w/o Clip	41.08	39.64	27.15	36.31	35.05	21.77	28.19	27.11	13.72	35.05	34.03	20.58
w/o TAU	32.99	32.76	20.34	22.36	19.99	7.61	15.29	13.27	4.83	21.75	19.26	7.06

Experiment

- *Zero-shot Evaluation*

Table 4: Results of zero-shot visual relation grounding

Methods	Acc_S	Acc_O	Acc_R
T-Rank V_1 [3]	4.05	4.08	1.37
T-Rank V_2 [3]	7.09	4.13	1.37
Co-occur [15]	<u>11.60</u>	<u>10.99</u>	<u>7.38</u>
vRGV (ours)	**18.94**	**17.23**	**10.27**

3

Deep Distance Transform for Tubular Structure Segmentation in CT Scans

CVPR 2020 oral

pdf

Task

- *Tubular Structure Segmentation in CT Scans*
  - Challenges: Poor contrast, noise and complicated background.

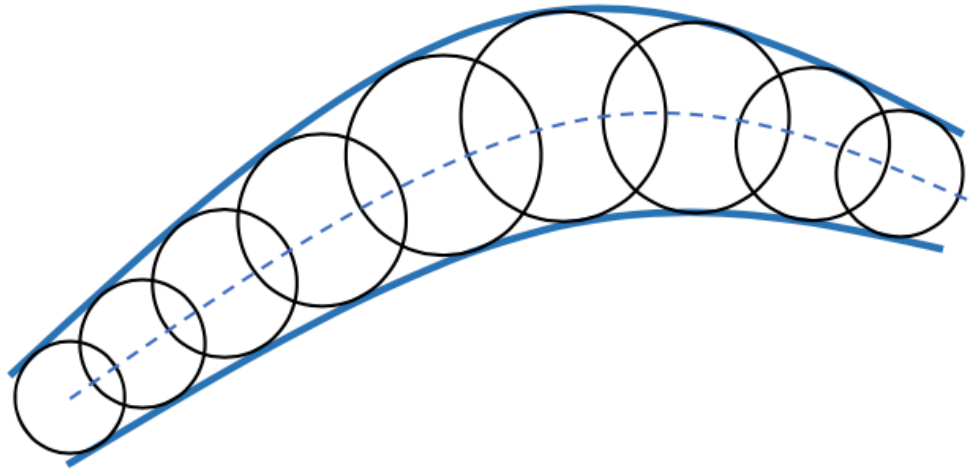


Figure 1. A tubular shape is presented as the envelope of a family of spheres with continuously changing center points and radii [9].

Motivation

- *Tubular Structure Segmentation in CT Scans*

  - Straightforward strategy:

    - Train a model, *e.g.*, a deep network, to directly predict whether each voxel is on the skeleton of the tubular structure or not as well as the cross-sectional radius of each skeleton point.
    - Reconstruct the segmentation of the tubular structure from its skeleton and radii.

  - Limitations:

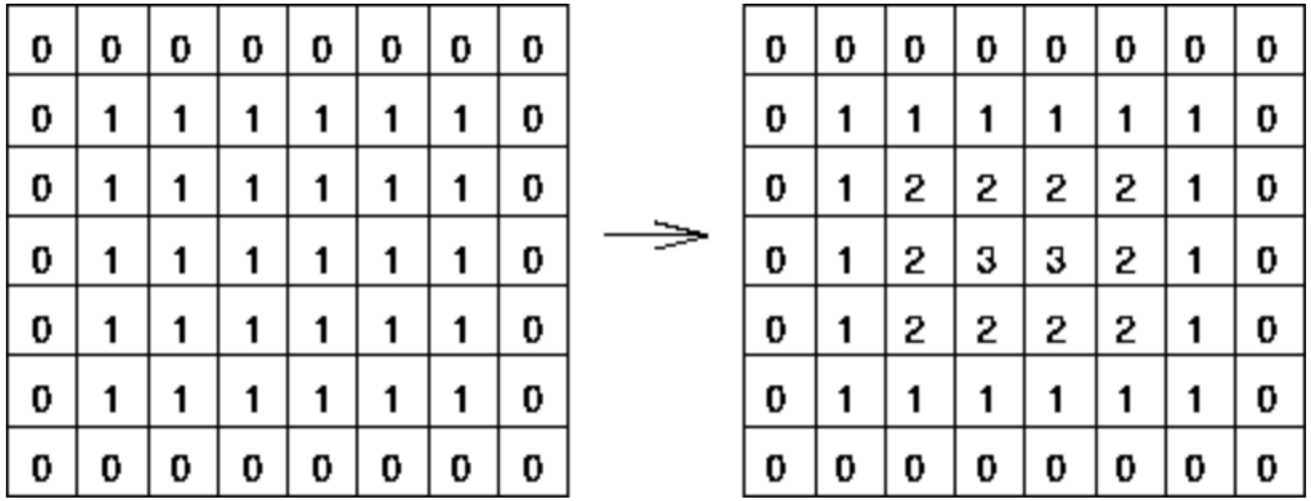
    - The ground-truth skeletons used for training are not easily obtained.
    - It is hard for the classifier to distinguish voxels on the skeleton itself from those immediately next to it, as they have similar features but different labels.

Method

- *Task definition*
  - Given a 3D CT scan X of size $(L \times W \times H)$, as a function on the coordinate set $V = \{v | v \in N^L \times N^W \times N^H\}$, *i.e.*, $X: V \to R \subset \mathbb{R}$ where the value on position v is defined as $x_v = X(v)$.
  - Goal: to predict the label $\hat{Y}$ of all voxels in the CT scan, where $\hat{y}_v \in \{0, 1\}$ denotes the predicted label for each voxel at position v, *i.e.*, if the voxel at v is defined as = X(v) is predicted as a tubular structure voxel, then $\hat{y}_v = 1$, otherwise $\hat{y}_v = 0$.

Method

- *Distance Transform*
  - The distance transform is an operator normally only applied to binary images. The result of the transform is a gray-level image that looks similar to the input image, except that the gray-level intensities of points inside foreground regions are changed to show the distance to the closest boundary from each point.



0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	2	2	2	2	1	0
0	1	2	3	3	2	1	0
0	1	2	2	2	2	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

Method

- *Distance Transform for Tubular Structure*

  - Given the ground-truth label map Y of the CT scan X in the training phase, let C_V be the set of voxels on the tubular structure surface,

$$C_V = \{\mathbf{v} |\, y_\mathbf{v} = 1, \exists\, \mathbf{u} \in \mathcal{N}(\mathbf{v}), y_\mathbf{u} = 0\}$$

    where $\mathcal{N}(\mathbf{v})$ denotes the 6-neighbour voxels of $\mathbf{v}$.

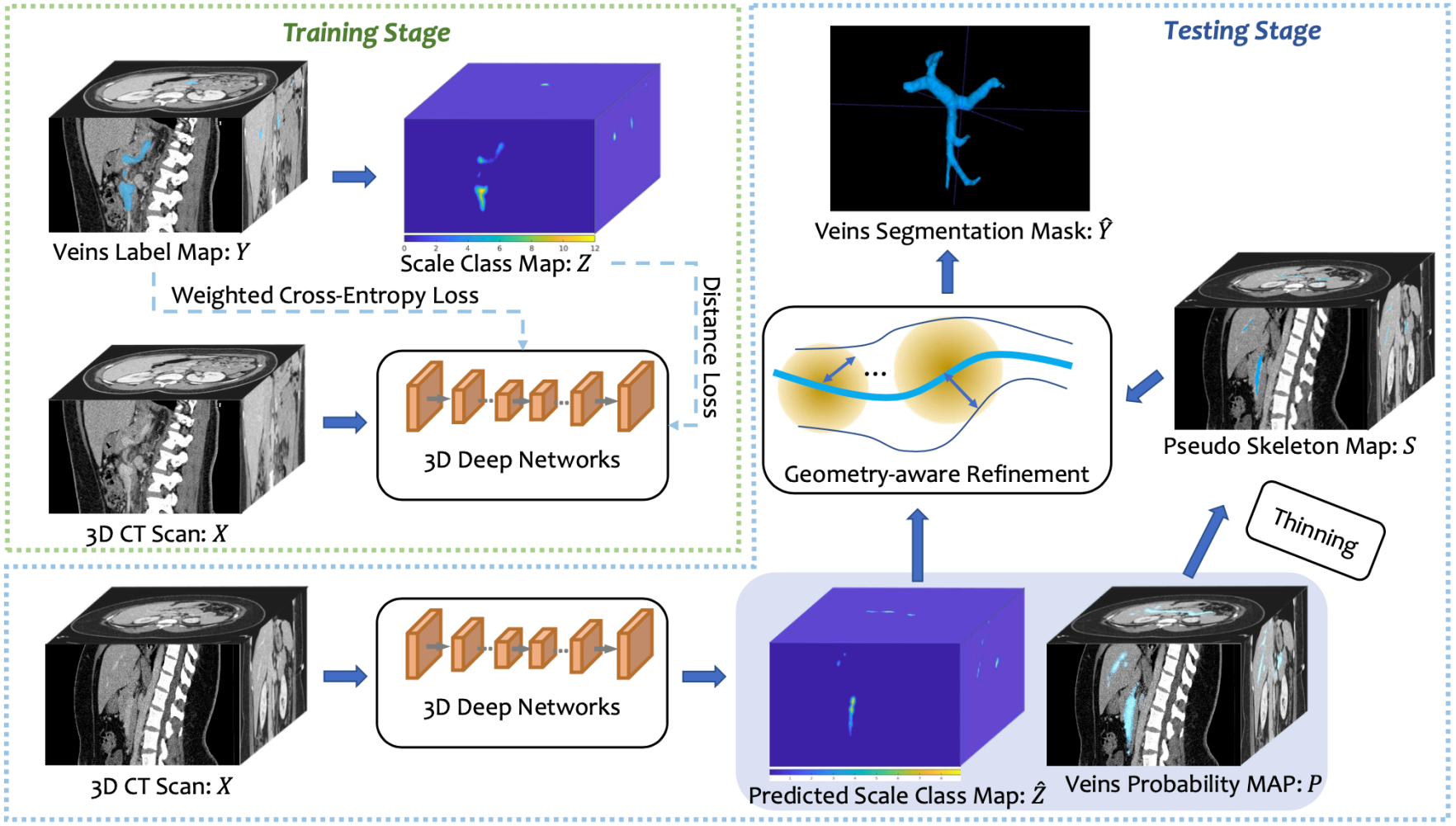
  - Distance Transform on the CT scan X: The distance map D is computed by

$$d_\mathbf{v} = \begin{cases} \min\limits_{\mathbf{u} \in C_V} \|\mathbf{v} - \mathbf{u}\|_2, & \text{if } y_\mathbf{v} = 1 \\ 0, & \text{if } y_\mathbf{v} = 0 \end{cases}.$$

  - Note that the values in the distance map D is real number, so each $d_\mathbf{v}$ is quantized into one of K bins by rounding it to the nearest integer, which converts the continuous distance map D to a discrete quantized distance map Z, where $z_\mathbf{v} \in \{0, \ldots, K\}$ (K-class classification).

Method

- *Network Training for Deep Distance Transform*



Method

- *Network Training for Deep Distance Transform*
  - The first branch is targeting on the ground-truth label map Y, which performs per-voxel classification for semantic segmentation with a weighted cross-entropy loss function $\mathcal{L}_{\text{cls}}$:

$$\mathcal{L}_{cls} = -\sum_{v \in V}(\beta_p y_v log\, p_v(W, w_{cls}) + \beta_n(1 - y_v) log\, (1 - p_v(W, w_{cls})))$$

where W is the parameters of the network backbone, w_{cls} is the parameters of this head branch and $p_v(W, w_{cls})$ is the probability that v is a tubular structure voxel as predicted by this head branch. $\beta_p = \frac{0.5}{\sum_{\mathbf{v}} y_{\mathbf{v}}}$ and $\beta_n = \frac{0.5}{\sum_{\mathbf{v}}(1-y_{\mathbf{v}})}$ are loss weights for tubular structure and background classes, respectively.

Method

- *Network Training for Deep Distance Transform*
  - The second head branch is predicting on the scale class map Z, which performs scale prediction for tubular structure voxels (*i.e.*, $z_{\mathbf{v}} > 0$). A new distance loss function $\mathcal{L}_{\text{dis}}$ to learn this head branch,

$$\mathcal{L}_{dis} = -\beta_p \sum_{v \in V} \sum_{k=1}^{K} \left(\mathbf{1}(z_v = k) \left(log\, g_v^k(W, w_{dis}) + \lambda \omega_v log\, (1 - \max_l g_v^l(W, w_{dis})) \right) \right)$$

where W is the parameters of the network backbone, w_{dis} is the parameters of the second head branch, $\mathbf{1}(\cdot)$ is an indication function, λ is a trade-off parameter which balances the two loss terms, $log\, g_v^k(W, w_{dis})$ is the probability that the scale of v belongs to k-th scale class and. $\omega_{\mathbf{v}}$ is a normalized weight defined by $\omega_{\mathbf{v}} = \frac{|\arg\max_l g_{\mathbf{v}}^l(\mathbf{W}, \mathbf{w}_{\text{dis}}) - z_{\mathbf{v}}|}{K}$.

Method

- *Geometry-aware Refinement*
  - **Pseudo skeleton generation**. The probability map P is thinned by thresholding it to generate a binary pseudo skeleton map S for the tubular structure. If $p_v > T^p$, $s_v = 1$; otherwise, $s_v = 0$,and T^p is the threshold.
  - **Shape reconstruction**. For each voxel $\mathbf{v}$, its predicted $\hat{z}_\mathbf{v}$ is given by $\hat{z}_\mathbf{v} = \arg\max_k g_\mathbf{v}^k$. A Gaussian kernel is fit to soften each sphere and obtain a soft reconstructed shape $\tilde{Y}^s$,

$$\tilde{y}_\mathbf{v}^s = \sum_{\mathbf{u} \in \{\mathbf{u}' | s_{\mathbf{u}'} > 0\}} c_\mathbf{u} \Phi(\mathbf{v}; \mathbf{u}, \mathbf{\Sigma}_\mathbf{u}),$$

where Σ_u is the co-variance matrix. According to the 3-sigma rule, $\Sigma_u = (\frac{\hat{z}_u}{3})^2$, where I is an identity matrix. The peak of $\Phi(\cdot; \mathbf{u}, \Sigma_u)$ becomes smaller if $\hat{z}_u$ is larger. A normalization factor $c_\mathbf{u} = \sqrt{(2\pi)^2 \det(\Sigma_u)}$ is introduced to normalize the peak of each distribution.

Method

- *Geometry-aware Refinement*
  - **Segmentation refinement**. The soft reconstructed shape $\tilde{Y}^s$ to refine the segmentation probability p_u, which produces a refined segmentation map $\tilde{Y}^r$,

$$\tilde{y}^r_{\mathbf{v}} = \sum_{\mathbf{u}\in\{\mathbf{u}'|s_{\mathbf{u}'}>0\}} p_{\mathbf{u}} c_{\mathbf{u}} \Phi(\mathbf{v}; \mathbf{u}, \mathbf{\Sigma}_{\mathbf{u}}).$$

The final segmentation mask $\hat{Y}$ is obtained by thresholding $\tilde{Y}^r$, *i.e.*, if $\tilde{y}^r_v > T_r$, $\hat{y}_v = 1$, otherwise,$\hat{y}_v = 0$, where $\tilde{y}^r_v$ and $\hat{y}_v$ are the value of voxel at position $\mathbf{v}$ of $\tilde{Y}^r$ and $\hat{Y}$, respectively.

Experiment

- *Results on PDAC Segmentation Dataset*

Table 1. Performance comparison (DSC, %) on pancreatic duct segmentation (mean ± standard deviation of all cases). SegBaseline stands for per-voxel classification. Multi-phase HPN is a hyper-paring network combining CT scans from both **venous** (V) and **arterial** (A) phases. Noted that only CT scans in **venous** phase are used for SegBaseline and DDT. **Bold** denotes the best results.

Methods	Phase	Backbone Networks	
		3D-UNet	ResDSN
SegBaseline [47]	V	40.25 ± 27.89	49.81 ± 26.23
Multi-phase HPN [47]	A+V	44.93 ± 24.88	**56.77** ± **23.33**
DDT (Ours)	V	**58.20** ± **23.39**	55.97 ± 24.76

Experiment

- *Results on PDAC Segmentation Dataset*

Table 2. Ablation study of pancreatic duct segmentation using ResDSN as backbone network. GAR indicates the proposed geometry-aware refinement.

Method	Average DSC (%)
SegBaseline [47]	49.81
SegfromSkel	51.88
DDT $\lambda = 0$, w/o GAR	52.73
DDT $\lambda = 0$, w/ GAR	54.70
DDT $\lambda = 1$, w/o GAR	53.69
DDT $\lambda = 1$, w/ GAR	**55.97**

Experiment

- *Results on Tubular Structure Datasets*

Table 3. Performance comparison (in average DSC, % and mean surface distance in mm) on three tubular structure datasets by using different backbones. "↑" and "↓" indicate the larger and the smaller the better, respectively. **Bold** denotes the best results for each tubular structure per measurement.

Backbone	Methods	Aorta		Veins		Pancreatic duct	
		Average DSC ↑	Mean Surface Distance ↓	Average DSC ↑	Mean Surface Distance ↓	Average DSC ↑	Mean surface Distance ↓
3D-HED [24]	SegBaseline	90.85	1.15	73.57	5.13	46.43	7.06
	DDT	92.94	0.82	76.20	**3.78**	54.43	4.91
3D-UNet [12]	SegBaseline	92.01	0.94	71.57	4.46	56.63	3.64
	DDT	**93.30**	**0.61**	75.59	4.07	**62.31**	**3.56**
ResDSN [50]	SegBaseline	89.89	1.12	71.10	6.25	55.91	4.24
	DDT	92.57	1.10	**76.60**	5.03	59.29	4.19

Experiment

- *Results on Hepatic Vessels Dataset in MSD Challenge*

Table 4. Comparison to competing submissions of MSD challenge: `http://medicaldecathlon.com`

Methods	Average DSC (%)
DDT (Ours)	**63.43**
nnU-Net [19]	63.00
UMCT [44]	63.00
K.A.V.athlon	62.00
LS Wang's Group	55.00
MIMI	60.00
MPUnet [26]	59.00

Experiment

- *Finding PDAC Tumor by Dilated Duct*

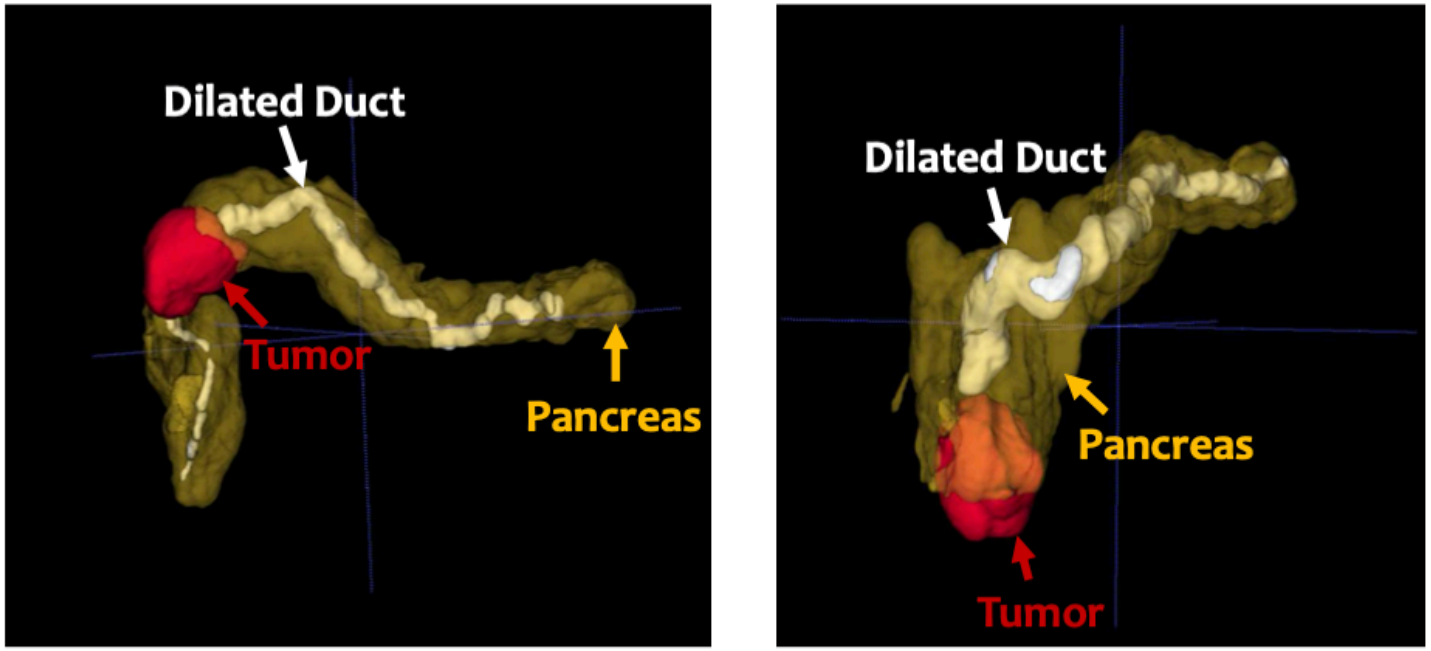


Figure 5. Examples of PDAC cases. In most PDAC cases, the tumor blocks the duct and causes it to dilate.

Experiment

- *Finding PDAC Tumor by Dilated Duct*

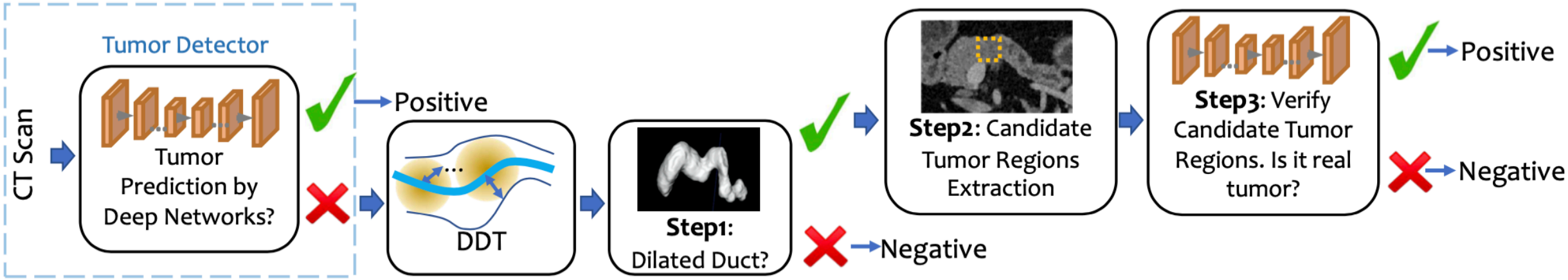


Figure 6. Flowchart of finding missing PDAC tumor by dilated duct.

Experiment

- *Finding PDAC Tumor bv Dilated Duct*

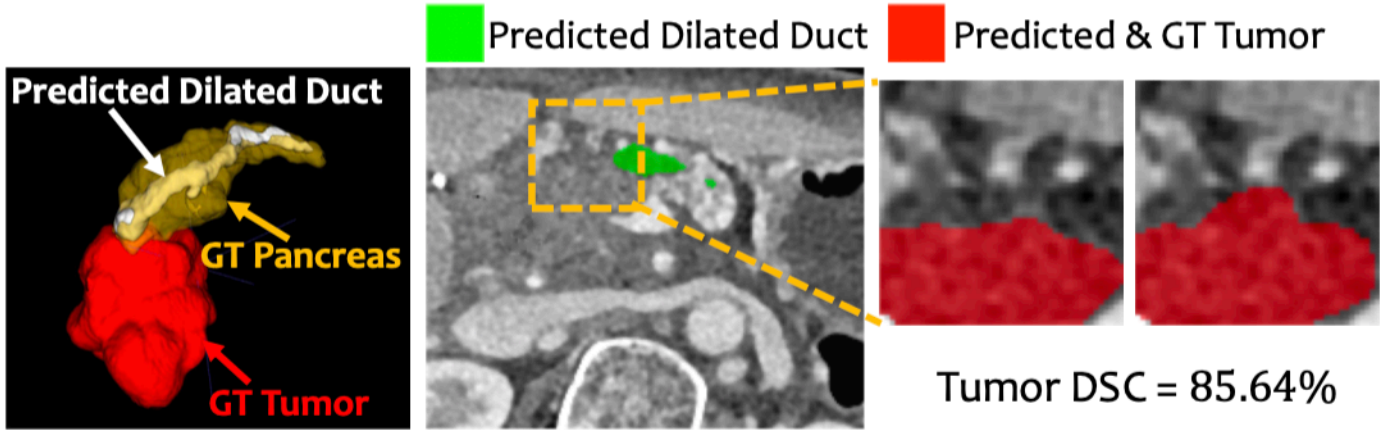


Figure 7. Examples of finding missed tumor of [51] by dilated duct. (a) The ground-truth tumor is right behind one end of the predicted dilated duct. The ground-truth pancreas is shown as a reference. (b) A cropped CT slice with predicted duct (we choose green for better visualization). The yellow dashed box is a candidate tumor region, shown in 2D. (c) and (d) are the same zoomed in image region with predicted and ground-truth tumor, respectively.

Thank You!