When Confidence Fails: Revisiting Pseudo-Label Selection in Semi-supervised Semantic Segmentation

Pan Liu, Jinshi Liu*

School of Automation，Central South University

CENTRAL SOUTH UNIVERSITY

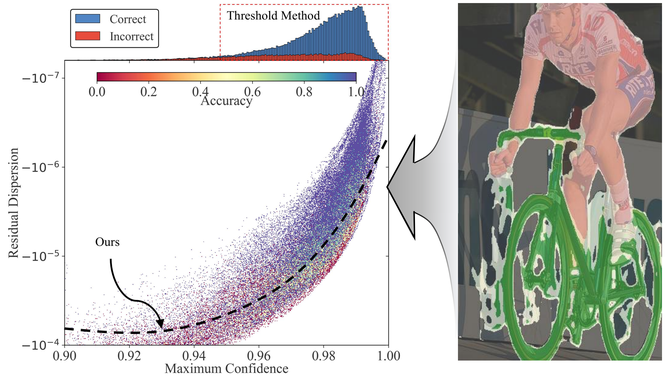
Project page

ICCV OCT 19-23, 2025 HONOLULU HAWAII

Problem

Motivation

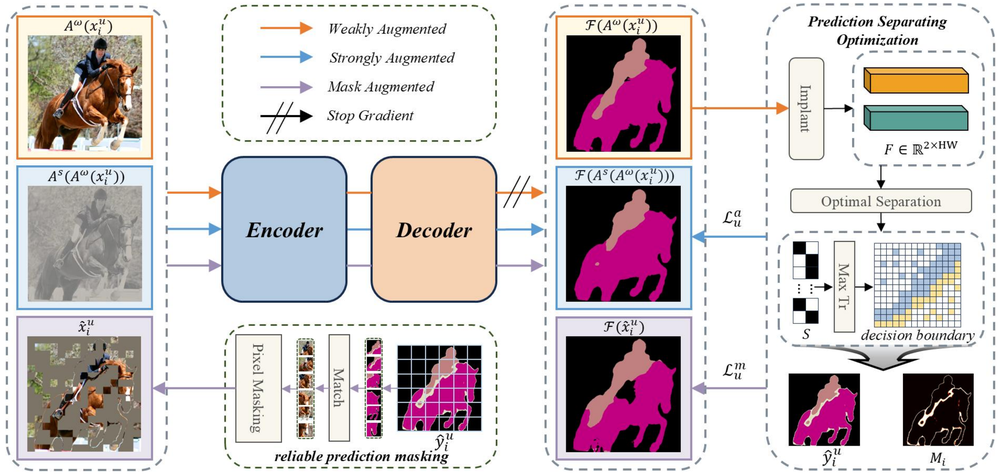
- Model confidence scores are poorly calibrated, failing to distinguish between correct and incorrect predictions.
- existing methods must discard low-confidence predictions, removes precisely the challenging and informative data.



Contribution

- Exposes limitations of max-confidence selection.
- A unified confidence feature space framework.
- Sample-adaptive pseudo-label selection.
- Leverages supervision from unreliable predictions.
- Achieves state-of-the-art benchmark performance.

Method



Proof

$$CE(p_n, q) \approx -\log p_n(k') - \frac{\epsilon(K-1)^3}{2(1-p_n(k'))^2} \cdot v_n$$

Pseudo-label selection benefits from both the maximum confidence and the residual confidence distribution.

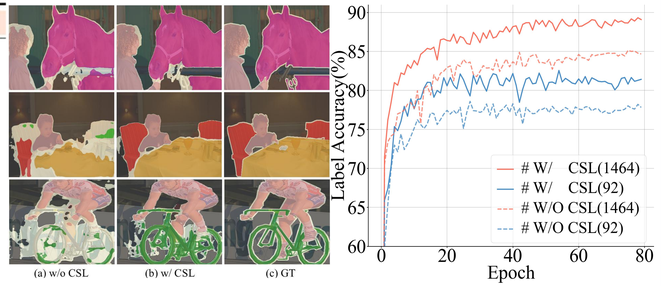
Selection Function

$$\max_{S} Tr\left(S^T \Phi^T \Phi S\right), s.t. S \in \{0,1\}^{HW \times 2}$$

Through a trace maximization objective, we achieve sample-adaptive selection of the most reliable predictions.

Experiments

PASCAL *original*	Train Size	1/16(92)	1/8(183)	1/4(366)	1/2(732)	Full(1464)
Supervised	321 × 321	45.4	53.1	61.2	68.4	73.2
ST++ [68]	321 × 321	65.2	71.0	74.6	77.3	79.1
UniMatch [69]	321 × 321	75.2	77.2	78.8	79.9	81.2
CorrMatch [50]	321 × 321	76.4	78.5	79.4	80.6	81.8
ESL [39]	513 × 513	71.0	74.1	78.1	79.5	81.8
LogicDiag [34]	513 × 513	73.3	76.7	78.0	79.4	-
RankMatch [40]	513 × 513	75.5	77.6	79.8	80.7	82.2
AugSeg [74]	513 × 513	71.1	75.5	78.8	80.3	81.4
SemiCVT [24]	513 × 513	68.6	71.3	75.0	78.5	80.3
FPL [44]	513 × 513	69.3	71.7	75.7	79.0	-
AllSpark [54]	513 × 513	76.1	78.4	79.8	80.8	82.1
CSL	321 × 321	**76.8**	**79.6**	**80.3**	**80.9**	**82.3**



More Pseudo-labels, Richer Supervision, Less Bias.

Background

Revisiting Confidence-Based Pseudo-Label Selection

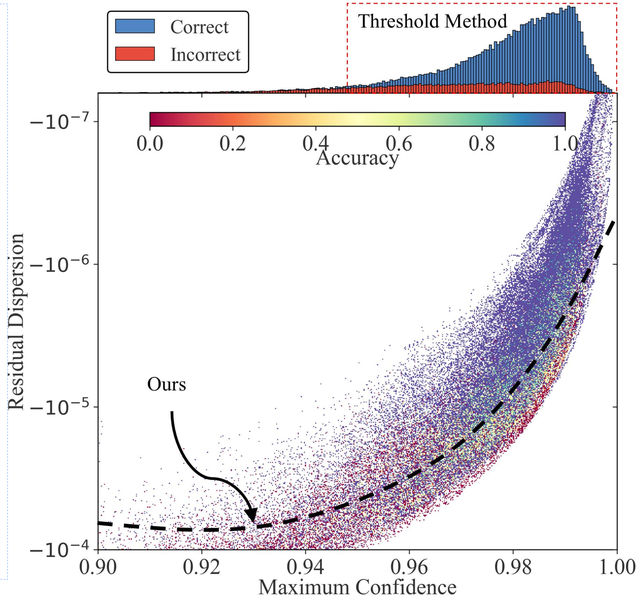
Challenges Lack of theoretical grounding for confidence-accuracy correlation, leading to model systemic overconfidence and suboptimal results.

Contribution First to prove that jointly optimizing max confidence and residual distribution disentangles overconfidence, enabling sample-adaptive pseudo-labeling.

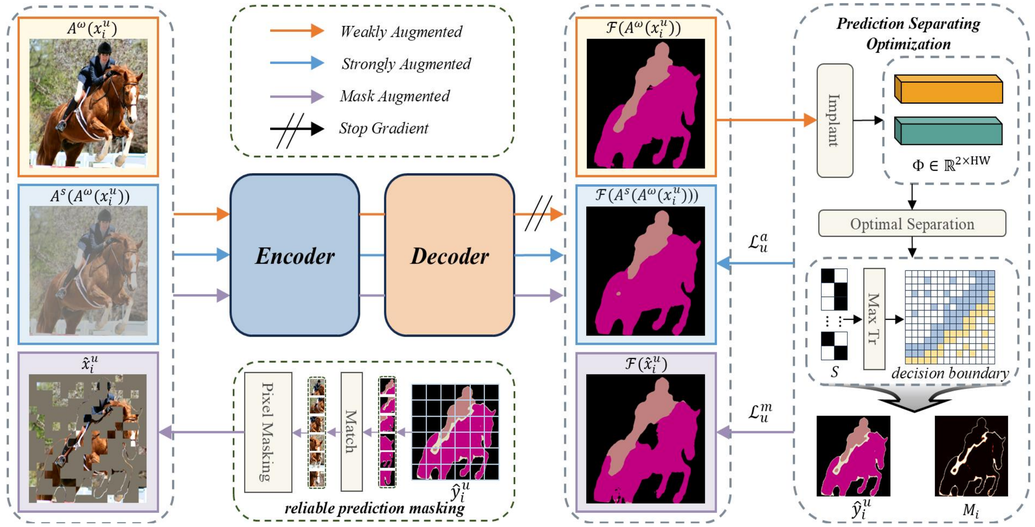
Challenges in SSSS

Confidence Collapse:
Predictions cluster at high confidence due to overconfidence.

Strict Filtering:
Results in insufficient supervision from limited labeled data.



Confidence Separable Learning (CSL)



Advances the statistical foundation of pseudo-label selection.

Confidence Separable Learning Framework

Sample-Wise Optimal Labeling via Spectral Optimization, Expanding Supervision with Trusted Masking

Sample-Adaptive Pseudo-Label Selection

Algorithm 1 Pseudocode of Prediction Convex Optimization Separation in a PyTorch-like style.

```
# pmax: Pixel-level maximum confidence
# vn: Pixel-level residual dispersion
def PCOS(pmax, vn):
  # combine pmax and vn into a feature matrix Φ
  Φ = stack([pmax, vn], axis=1).T
  # extract the top two eigenvectors from Φ
  U, Sigma, VT = svd(Φ)
  eig_vectors = VT[:, :2]
  # constructing the optimal selection matrix
  S = argmax(abs(eig_vectors), axis=1)
  # calculate stats for each class
  stats = [(Φ[S == c].mean(dim=1),
     Φ[S == c].std(dim=1)) for c in range(2)]
  # select the reliable class
  mu, sigma = max(stats, key=lambda x:x[0])
  # smooth loss weight
  weight = exp(-((Φ-mu)/(8*sigma))**2)
  weight = weight.prod(dim=0)
  # preserving reliable prediction weights
  weight[(Φ[0, :]>mu[0])‖(Φ[1, :]>mu[1])] = 1
  return weight
```

$$\max_{\mathbf{S}} Tr\left(S^T\Phi^T\Phi S\right), s.t. S \in \{0,1\}^{HW\times 2}$$

$$S^*_{n,c} = \Delta\left(\arg\max_{i\in\{1,2\}} |u_i(n)|, c\right)$$

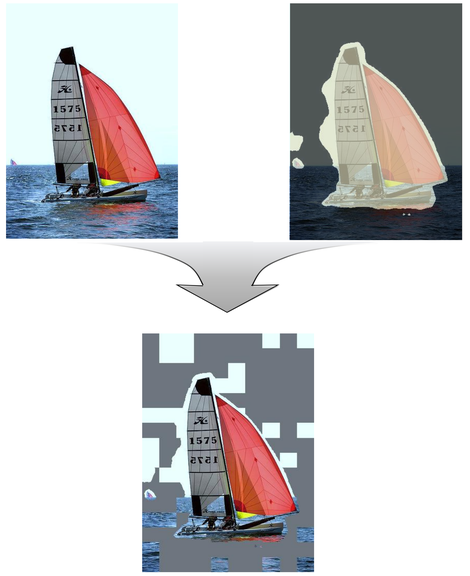
$$\omega_n = \prod_c \exp\left(\frac{(h_n(c)-\mu_c)^2}{-\alpha\sigma_c^2}\right)$$

$$M_i(n) = \begin{cases} 1 & \text{if } (h_n(c)-\mu_c) > 0, \forall c \in \{1,2\} \\ \omega_n & \text{otherwise} \end{cases}$$

Trusted Masking Strategy

Algorithm 2 Pseudocode of Trusted Mask Perturbation in a PyTorch-like style.

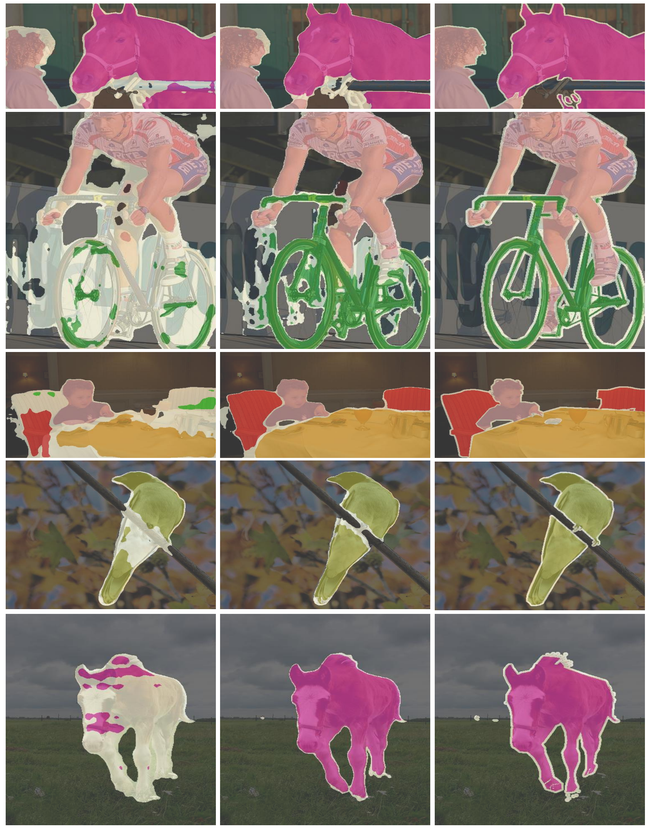
```
# x_w: Image with weak augmentation perturbation
# image_size: The length or width of the image
# block_size: The masking patch size
# masking_rate: The masking pixel ratio
# f: segmentation network
pred_w = f(x_w)
mask_w = pred_w.argmax(dim=1).detach()
# compute weights using PCOS on the projection
weight = PCOS(Projection(pred_w))
# create a reliability mask
reli_mask = (weight == 1)
# gain patch-based perturbation mask (Eq. (12))
mask_size = img_size // block_size
cover_mask = (rand(mask_size, mask_size) <
              masking_rate).float()
cover_mask = interpolate(cover_mask,
             size=img_size, mode='nearest')
# perturbation only for reliable predictions
cover_mask = cover_mask & reli_mask
# constructing perturbed images
x_m = x_s.clone()
x_m[cover_mask == 1] = 0
pred_m = f(x_m)
# calculated loss
criterion = CrossEntropyLoss()
loss_m = criterion(pred_m, mask_w)
```



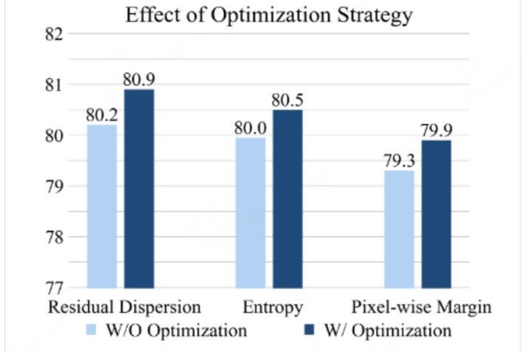
Experiment Results

Comparison, ablation experiments, and visualization results

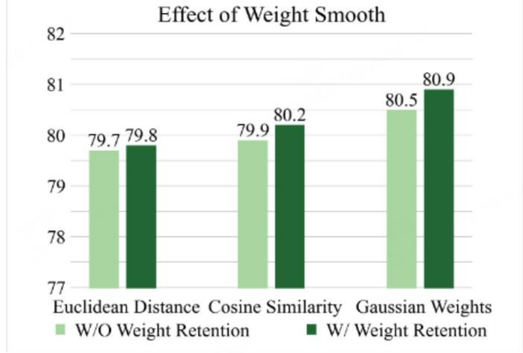
Problem



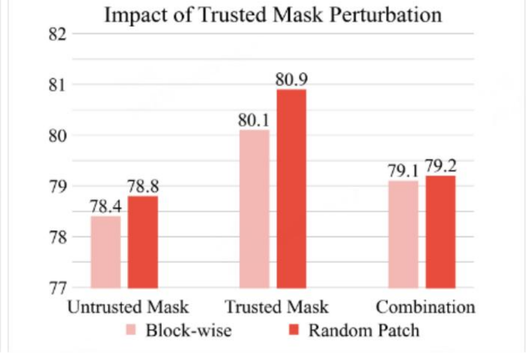
Problem



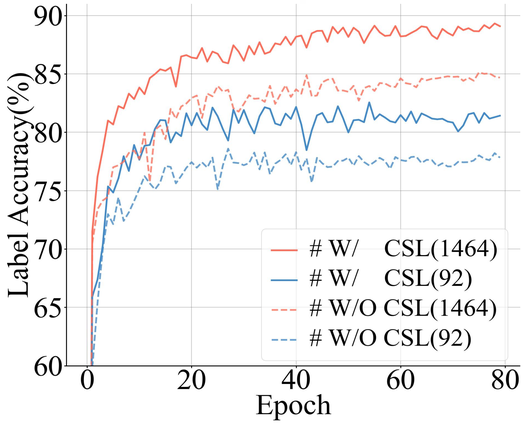
(a) Dispersion Metrics

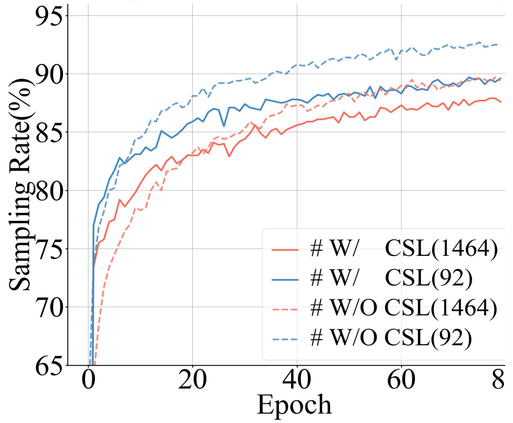


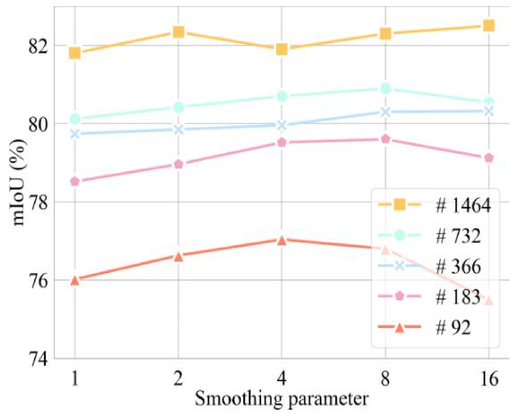
(b) Loss Smoothing Strategy



(c) Trusted Mask Perturbation







The effectiveness of CSL