Supplementary Material

More visual comparisons are provided as suggested by reviewer 3. In addition, we provided the pseudo-code of our proposed PPS is shown in Algorithm 1. The testing codes, the trained model and all results using our method can be found at `https://github.com/zxg3017/PPS`.

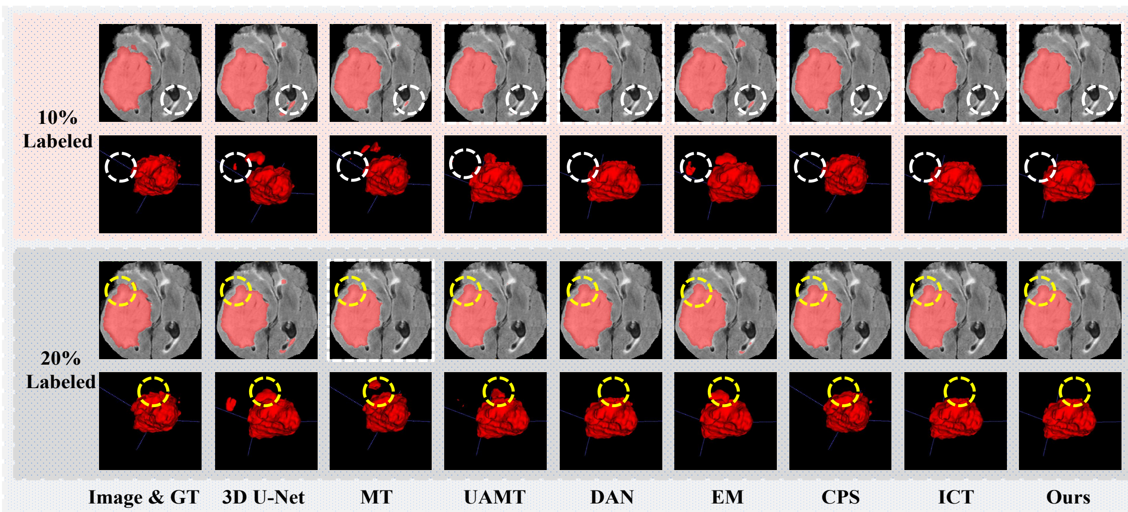
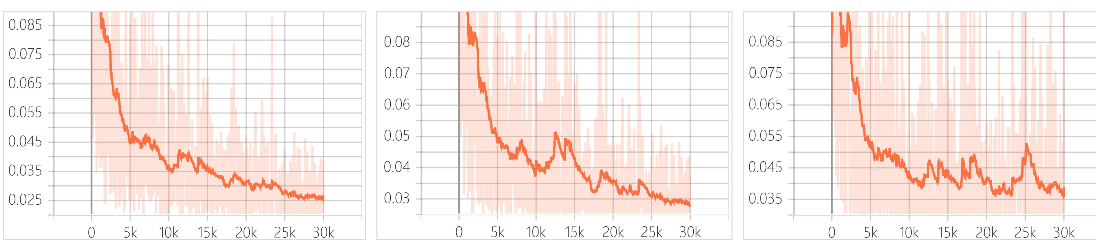


Figure 1: Visualization of brain tumor segmentation performance produced from several CNN models on the BraTS2019 dataset.



(a) **(b)** **(c)**

Figure 2: Smoothened loss of training data for our model A, model B, and model C on the BraTS2019 dataset.

Algorithm 1: Our proposed PPS's pseudo code.

Input: Batch of labeled volume and their one-hot labels $X^l = (X_b^l, X_b^y)$, batch of unlabeled samples $X^u = (X_b^u)$.

Output: PPS model's parameter

Initialization: Initialization parameters of $\theta_i, \{i \in [1, 2, 3]\}$perturbation of Gaussian noise $\xi^i, i \in [1, 2, 3]$, learning rate (LR), and maximum iterations N

for *n = 1 to N* **do**

for *b = 1 to m* **do**

$(\hat{X}_b^l, \hat{X}_b^u) = WeakAug(X_b^l, X_b^u)$ //Apply weak augmentation to X

$f(X;\theta_1) \rightarrow Y_1 \rightarrow P_1$ // Calculate the segmentation confidence map Y_1 and the pseudo label P_1 by Eq. (1).

$f(X;\theta_2) \rightarrow Y_2 \rightarrow P_2$ // Calculate the segmentation confidence map Y_2 and the pseudo label P_2 by Eq. (2).

$f(X;\theta_3) \rightarrow Y_3 \rightarrow P_3$ // Calculate the segmentation confidence map Y_3 and the pseudo label P_3 by Eq. (3).

for *each model in* $f(X;\theta_1), f(X;\theta_1), f(X;\theta_1)$ **do**

Calculate $\mathcal{L}_{sup}$ on labeled data by Eq. (5)

Calculate $\mathcal{L}_{pps}$ on unlabeled data by Eq. (6)

Update $f(X;\theta_1), f(X;\theta_2), f(X;\theta_3)$ by the final loss $\mathcal{L}_{total}$ in Eq. (7)