A More Empirical Studies for Section 2.1

We conduct the same empirical studies on CIFAR-10 dataset with the selection metrics computed at various epochs. We show epoch 50 results in Figures 6, 7 and 8 and epoch 200 results in Figures 9, 10 and 11. We observe similar patterns as discussed in Section 2.1

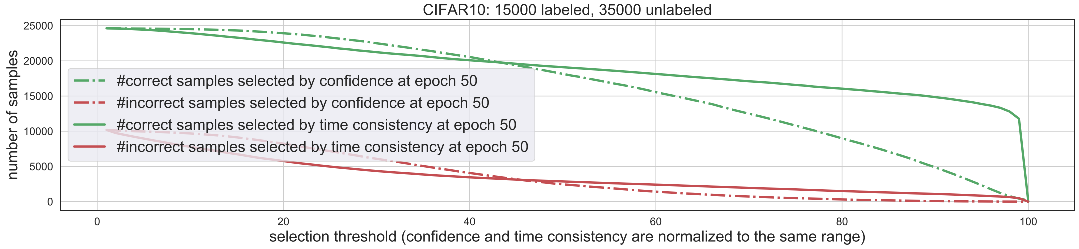

Figure 6. Compute time-consistency and confidence at epoch 50. Select samples in the validation set based on different thresholds of the two metrics and report the number of correct v.s. incorrect predictions in the selected samples. The two metrics are normalized to $[0, 1]$ range.

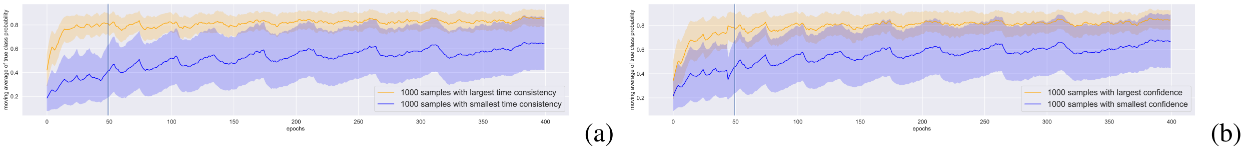

Figure 7. Compute time-consistency (a) and confidence (b) at epoch 50. Select the top 1000 and bottom 1000 samples in the validation set based on the two metrics. Compare the moving average of true class probability of the selected samples across training epochs.

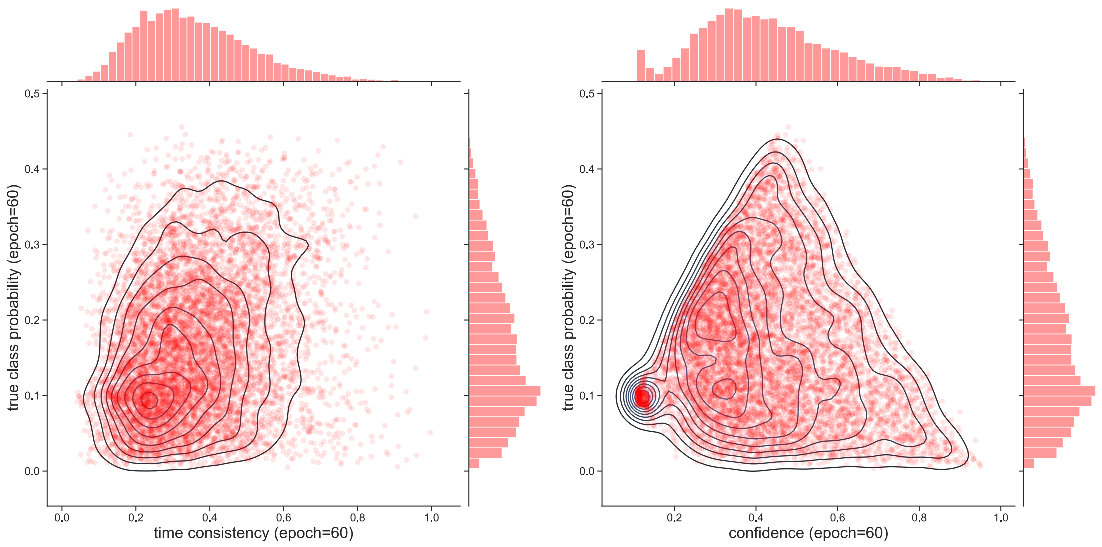

Figure 8. Compute time-consistency and confidence at epoch 60. Plot the incorrectly predicted samples in the validation set as a scatter plot with the selection metric as the x-axis and the true class probability as the y-axis.

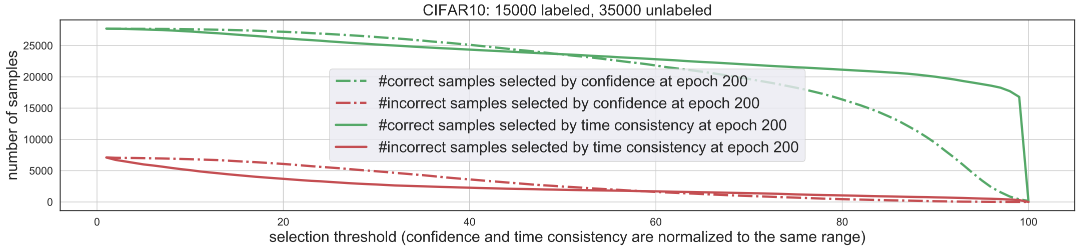

Figure 9. Compute time-consistency and confidence at epoch 200. Select samples in the validation set based on different thresholds of the two metrics and report the number of correct v.s. incorrect predictions in the selected samples. The two metrics are normalized to $[0, 1]$ range.

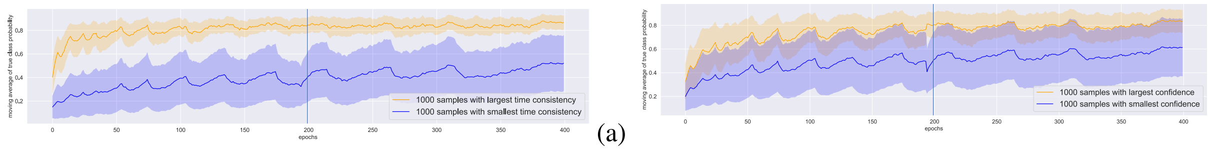

Figure 10. Compute time-consistency (a) and confidence (b) at epoch 200. Select the top 1000 and bottom 1000 samples in the validation set based on the two metrics. Compare the moving average of true class probability of the selected samples across training epochs.

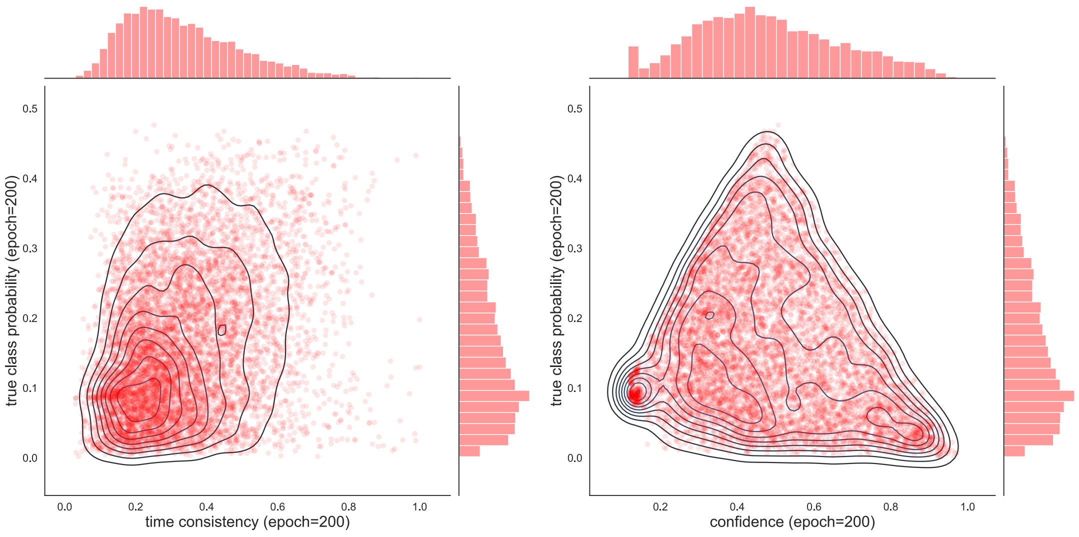

Figure 11. Compute time-consistency and confidence at epoch 200. Plot the incorrectly predicted samples in the validation set as a scatter plot with the selection metric as the x-axis and the true class probability as the y-axis.

B Details on Eq. (5) deriving Time Consistency

Let $z = y^t(x')$, we have:

$$
\begin{aligned}
&\left|(\boldsymbol{y}^t(x') - p^t(x'))\frac{\partial f^t(x')}{\partial t}\right| && (12)\\
\approx &\left|(\boldsymbol{y}^t(x') - p^t(x'))(f^{t+1}(x') - f^t(x'))\right| && (13)\\
= &\left|(1 - p_z^t(x'))(f_z^{t+1}(x') - f_z^t(x')) - \sum_{j \neq z} p_j^t(x')(f_j^{t+1}(x') - f_j^t(x'))\right| && (14)\\
= &\left|- \left[\sum_j p_j^t(f_j^{t+1}(x') - f_j^t(x'))\right] + (f_z^{t+1}(x') - f_z^t(x'))\right| && (15)\\
= &\left|\left[- \sum_j p_j^t(x') \log \frac{\exp(f_j^{t+1}(x'))}{\exp(f_j^t(x'))} - p_j^t(x') \log \frac{\sum_i \exp(f_i^t(x'))}{\sum_i \exp(f_i^{t+1}(x'))}\right]\right.\\
&\left. + \log \frac{\sum_i \exp(f_i^t(x'))}{\sum_i \exp(f_i^{t+1}(x'))} \sum_j p_j^t(x') + (f_z^{t+1}(x') - f_z^t(x'))\right| && (16)\\
= &\left|D_{KL}(p^t(x')||p^{t+1}(x')) + \log \frac{\sum_i \exp(f_i^t(x'))}{\sum_i \exp(f_i^{t+1}(x'))} + (f_z^{t+1}(x') - f_z^t(x'))\right| && (17)\\
= &\left|D_{KL}(p^t(x')||p^{t+1}(x')) + (\log \sum_i \exp(f_i^t(x')) - \log \exp(f_z^t(x')))\right.\\
&\left. +(\log \exp(f_z^{t+1}(x')) - \sum_i \exp(f_i^{t+1}(x')))\right| && (18)\\
= &\left|D_{KL}(p^t(x')||p^{t+1}(x')) + \log \frac{p_z^{t+1}(x')}{p_z^t(x')}\right| && (19)\\
\leq &D_{KL}(p^t(x')||p^{t+1}(x')) + \left|\log \frac{p_z^{t+1}(x')}{p_z^t(x')}\right|. && (20)
\end{aligned}
$$

C More Experimental Details

C.1 Implementation

We run Pytorch re-implementation of MixMatch at `https://github.com/YU1ut/MixMatch-pytorch` to achieve MixMatch's results of training large WideResNet-28-135 on CIFAR10 and CIFAR100. All the other baselines' results are from MixMatch and ReMixMatch paper (Berthelot et al., 2019; 2020). Due to limited computation resources, we cannot run experiments for all the baselines and any missing results are marked by "-".

We run our Pytorch implementation of TC-SSL for each experiment on one NVIDIA GeForce RTX2080Ti or TESLA V100.

C.2 Hyperparameters

We do not heavily tune the hyperparameters for computational reasons. We tested limited options for $\gamma_\theta, \gamma_c, \lambda_{ct}, \lambda_{cs}$ and chose the ones with greatest improvement on validation set after 5 episodes. We only tune them on CIFAR10 with 500 labeled samples, and use the same hyperparameters for all other experiments.

The hyperparameters in TC-SSL can be divided into three groups: epochs T_0 and T, regularization weights λs, and multiplicative factors γs:

(1) For epochs, we fix warm starting epochs as $T_0 = 10$ since 10 epochs of training usually results in a reasonable model to start with; we limit the total epochs $T = 680$ due to limited computation.

(2) For γs, we fix $gamma_k = 0.005$ so the last episode is trained on all unlabeled data, i.e., $k = |U|$. We tried several common discounting factors for $\gamma_\theta, \gamma_c = 0.995, 0.99, 0.98$.

(3) For λs, we fixed $\gamma_{ce} = 1$ so the selected unlabeled and labeled samples have equal weights of classification loss in the objective. In tuning, we tried $\lambda_{ct} = 0.2, 0.4, 0.8$ and $\lambda_{cs} = 10/C, 20/C, 40/C$.

C.3 Test Accuracy during Training (MixMatch vs. TCSSL)

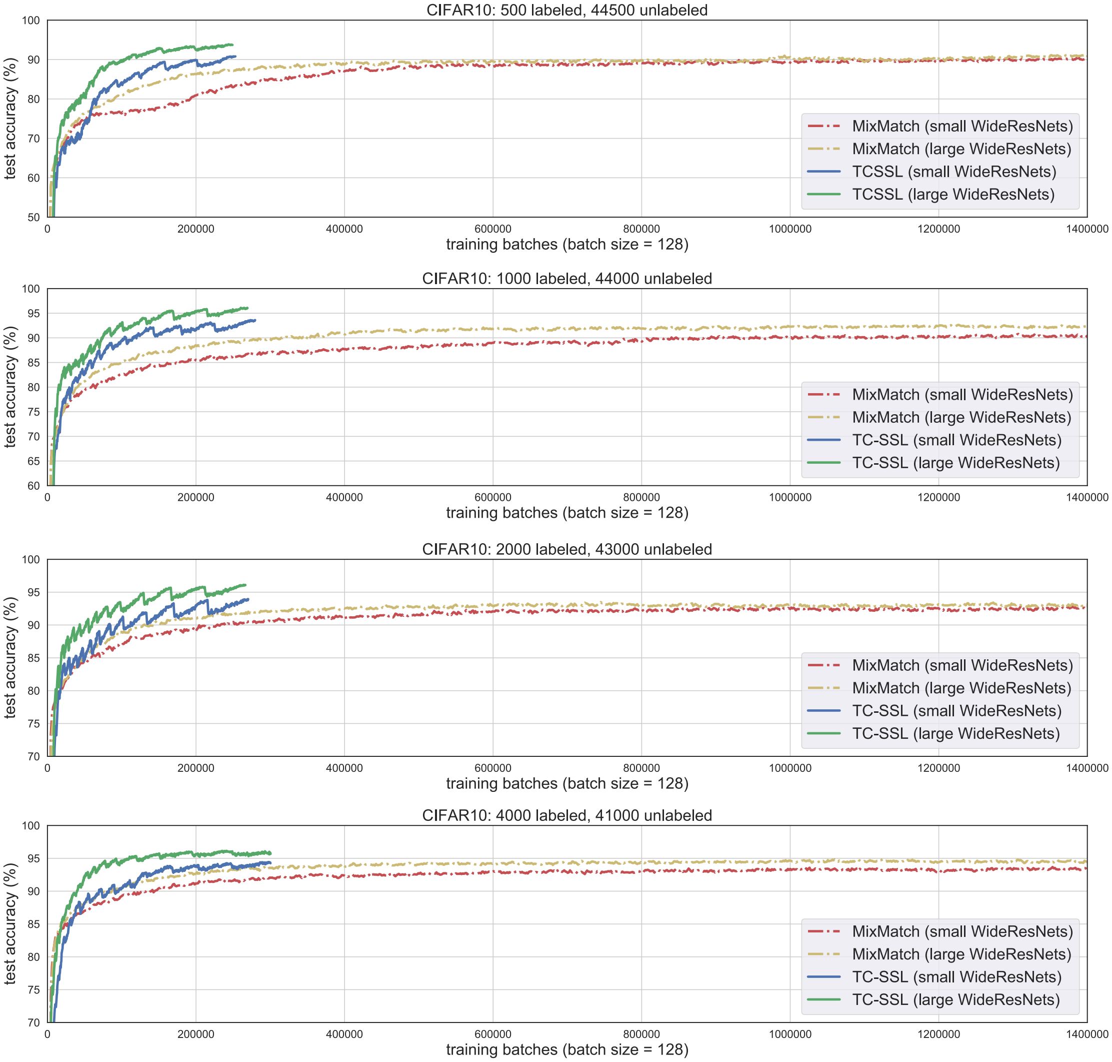

Figure 12. Test accuracy (%) during the training of small WideResNet and large WideResNet by MixMatch and TC-SSL on the four splittings of CIFAR10.

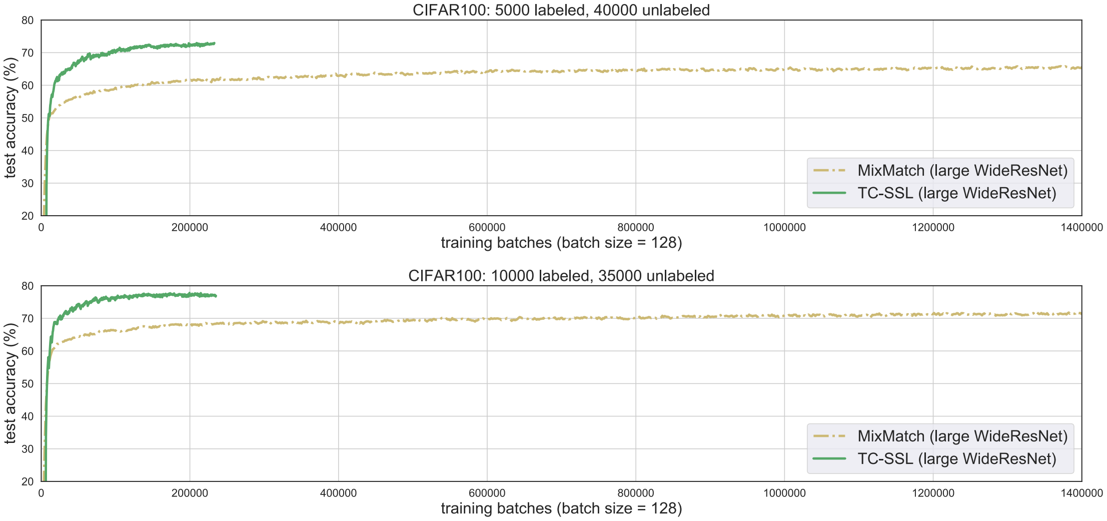

Figure 13. Test accuracy (%) during the training of the large WideResNet by MixMatch and TC-SSL on two splittings of CIFAR100.