Time-Consistent Self-Supervision for Semi-Supervised Learning

Tianyi Zhou*, Shengjie Wang*, Jeff A. Bilmes

University *of* Washington, Seattle

MELODI

MachinE Learning, Optimization, & Data Interpretation @ UW

PAUL G. ALLEN SCHOOL OF COMPUTER SCIENCE & ENGINEERING

ELECTRICAL & COMPUTER ENGINEERING

UNIVERSITY *of* WASHINGTON

UNIVERSITY OF WASHINGTON 1861 LVX SIT

Can SSL achieve fully-supervision's accuracy using similar amount of computation?

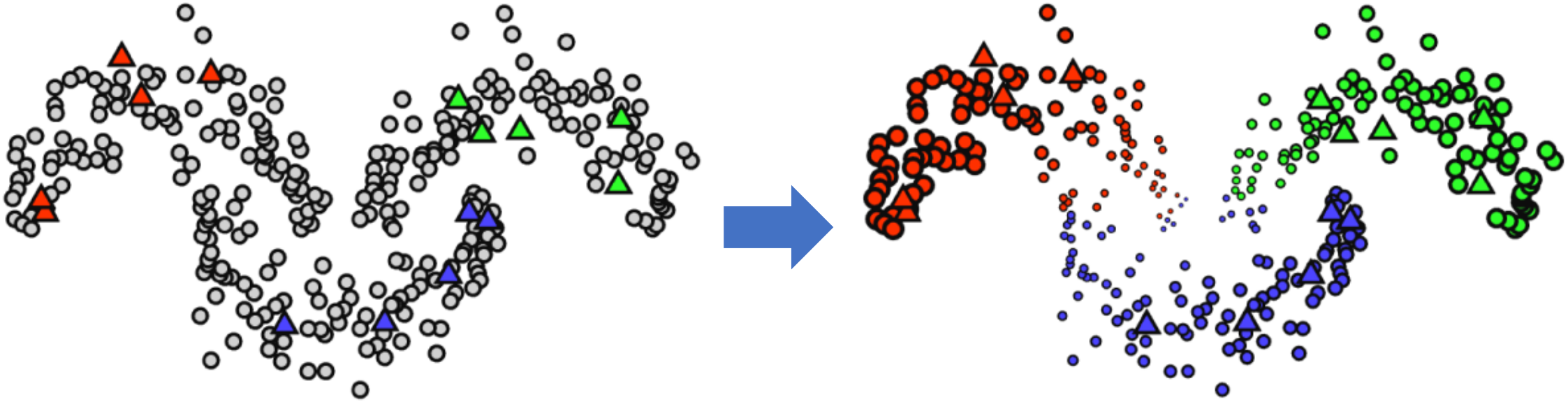
Yes!

How?

Select unlabeled data with time-consistent prediction for self-supervision.

Semi-Supervised Learning with **Spatial Consistency**

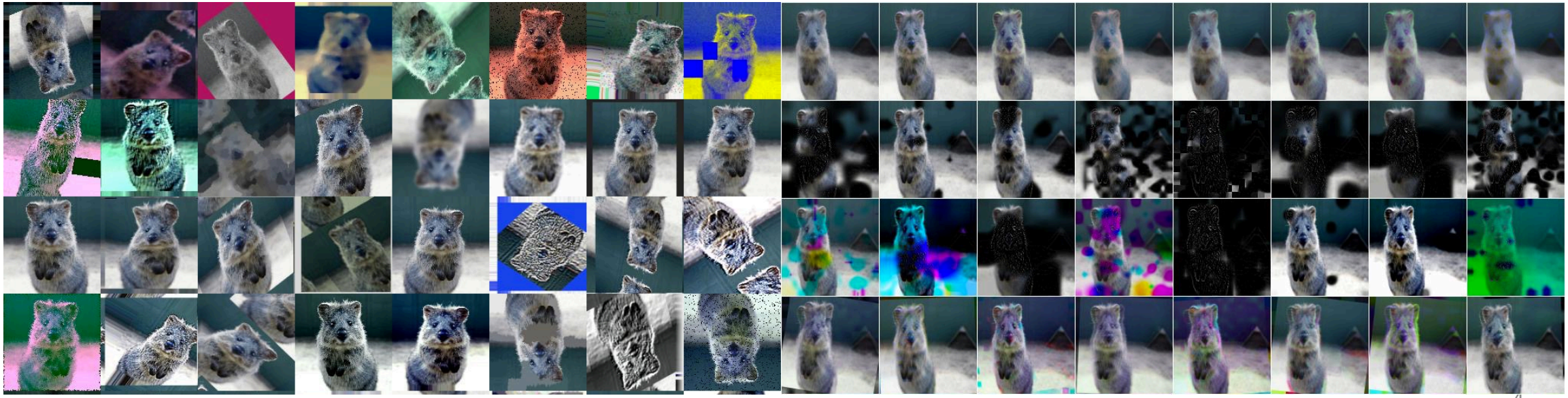
- *Idea:* Samples with similar features/embeddings have similar labels
- *Previous:* Label/measure propagation; manifold regularization
- *More recent:* The same idea inspires graphical neural networks



credit: [Iscen et al. 2019]

Semi-Supervised Learning with **Pseudo Targets**

- *Idea*: average the model output of an unlabeled sample over multiple augmentations/steps; use the average as training target.
- *Fit in Deep learning*: encourage spatial consistency around single sample; working with data augmentation and inductive bias of DNNs.
- *Drawbacks:* can be wrong on some samples; early-stage model is poor
  - In practice: select samples with high confidence, but DNNs can be over-confident.



credit: https://github.com/aleju/imgaug

A Recipe of **Self-Supervision** on Unlabeled Data

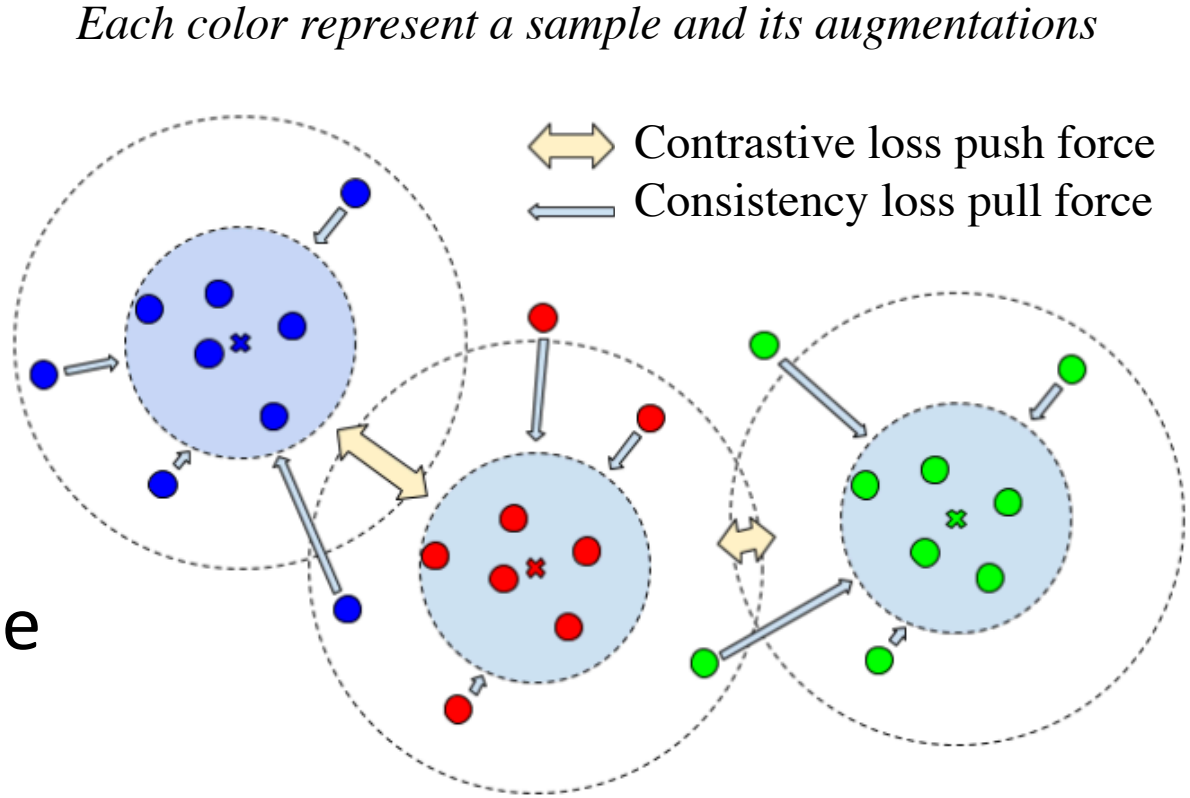
- *Consistency loss:*

  An unlabeled sample and its augmentation should have **similar predictions**.

- *Contrastive loss/Triplet loss:*

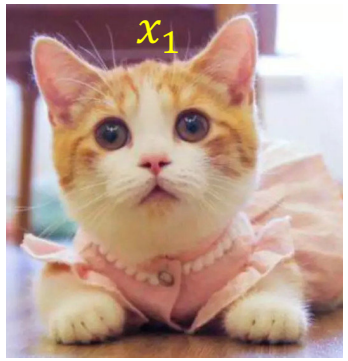
  Different samples (and their augmentations) should have **more different predictions** than the same sample and its augmentations.

- *Cross Entropy loss* defined on **pseudo targets**.
- *Our SSL objective combines the three losses.*



credit: [Brabandere et al. 2017]

An Example of Consistency/Contrastive Loss



CNN $f(\cdot)$

Shared weights

CNN $f(\cdot)$

Shared weights

CNN $f(\cdot)$

$f(x_1)$

$f(x_2)$

$f(x_3)$

Consistency loss

Contrastive Triplet Loss

$$\min_{f(\cdot)} \|f(x_1) - f(x_2)\|_p$$

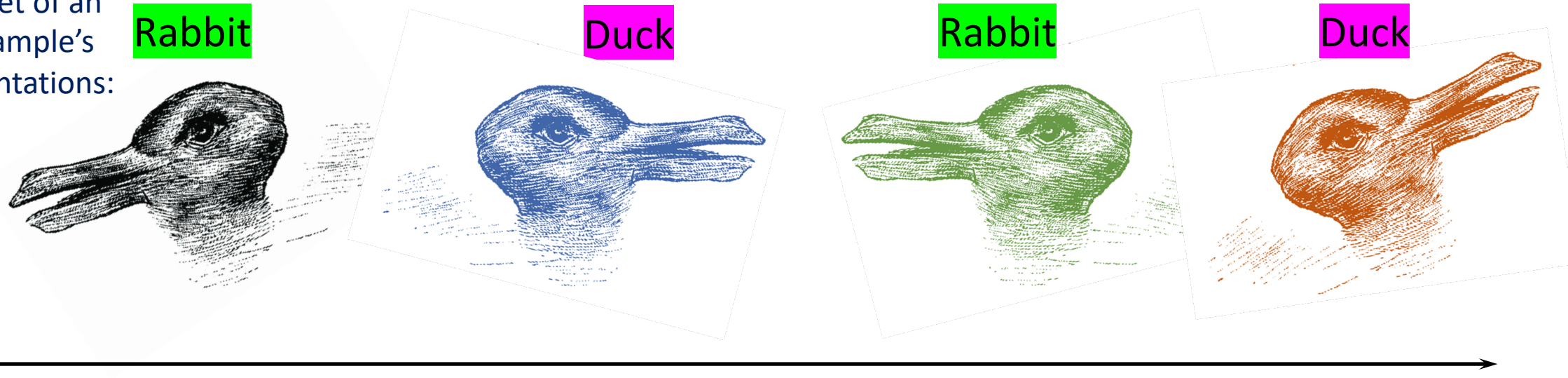
$$\min_{f(\cdot)} -\log \frac{\exp[\cos(f(x_1), f(x_2))]}{\sum_x \exp[\cos(f(x_1), f(x))]}$$

credit: https://mc.ai/face-recognition-using-one-shot-learning/

Problem of Current SSL: time-inconsistency

- The pseudo target depends on model in-training and is time-variant.
- Hence, the training objective is time-inconsistent!
- DNN is confusing itself in self-supervision.
- *Possible outcomes*: divergence, concept drift, catastrophic forgetting, etc.

Pseudo Target of an unlabeled sample's data augmentations:



Training Time

Self-supervision losses depend on pseudo targets (or model outputs), which should be time-consistent!

Time-Consistency (TC)

- We select unlabeled data with consistent predictions/outputs for self-supervision in SSL by using a curriculum.

- (instantaneous) Time consistency of sample x at step-t (e.g., t^{th} mini-batch):

$$a^t(x) \triangleq D_{KL}(p^{t-1}(x)||p^t(x)) + \left|\log \frac{p^{t-1}(x)[y^{t-1}(x)]}{p^t(x)[y^{t-1}(x)]}\right|$$

$p^t(x)$: output distribution over classes for x at step-t
$y^t(x)$: predicted class for x at step-t

Time-Consistency (TC)

- (instantaneous) Time consistency of x at step-t:

$$a^t(x) \triangleq D_{KL}(p^{t-1}(x)||p^t(x)) + \left|\log \frac{p^{t-1}(x)[y^{t-1}(x)]}{p^t(x)[y^{t-1}(x)]}\right|$$

  - $y^t(x) = \arg\max_i p^t(x)[i]$, i. e., the class with the highest probability.

- 1st term: *KL-divergence* between the predictions at step t and t-1.
- 2nd term: change of confidence on the predicted class between step t and $t-1$.

Time-Consistency (TC)

- (instantaneous) Time consistency of x at step-t:

$$a^t(x) \triangleq D_{KL}(p^{t-1}(x)||p^t(x)) + \left|\log \frac{p^{t-1}(x)[y^{t-1}(x)]}{p^t(x)[y^{t-1}(x)]}\right|$$

  - $y^t(x) = \arg\max_i p^t(x)[i]$, i. e., the class with the highest probability.

- 1st term: *KL-divergence* between the predictions at step t and t-1.
- 2nd term: change of confidence on the predicted class between step t and $t-1$.

Time-Consistency (TC)

- (instantaneous) Time consistency of x at step-t:

$$a^t(x) \triangleq D_{KL}(p^{t-1}(x)||p^t(x)) + \left|\log \frac{p^{t-1}(x)[y^{t-1}(x)]}{p^t(x)[y^{t-1}(x)]}\right|$$

- Time Consistency (TC): smooth $-a^t(x)$ by exponential moving average over time steps:

$$c^t(x) = \gamma_c(-a^t(x)) + (1-\gamma_c)c^{t-1}(x)$$

Time-Consistency relates to **Catastrophic Forgetting** in **Training Dynamics**

- $a^t(x')$ is an upper-bound on the forgetfulness of catastrophic forgetting on labeled data if adding an unlabeled sampel x' and its pseudo targets to training:

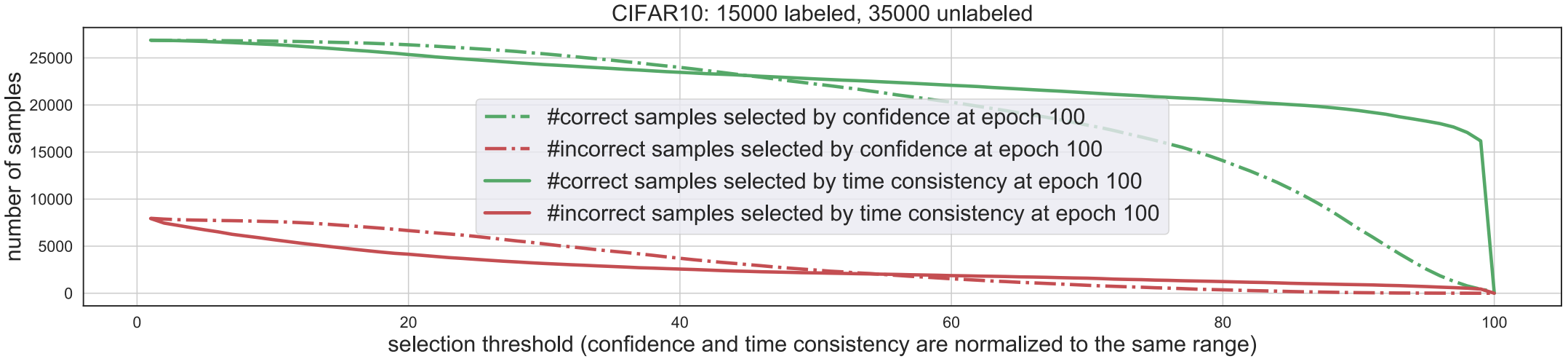
$$Forgetfulness \triangleq \frac{1}{\eta}\left|\sum_{x\in\mathcal{L}}\left[\ell(x;\theta^{t+1}) - \ell(x;\hat{\theta}^{t+1})\right]\right|$$

  - $\ell(x;\theta)$: loss of model θ on sample x;
  - Assume the loss on labeled data L is close to 0 after warm-starting epochs, i.e., $\sum_{x\in L}\ell(x;\theta^t)\approx 0$.
  - θ^t: model-at-step-t updated by labeled data;
  - $\hat{\theta}^t$: model-at-step-t updated by labeled data + x';

- A small $a^t(x')$ means adding x' and its pseudo target to training does not cause forgetting of labeled data (and previously trained unlabeled-data).

Empirical Evidence of Time Consistency

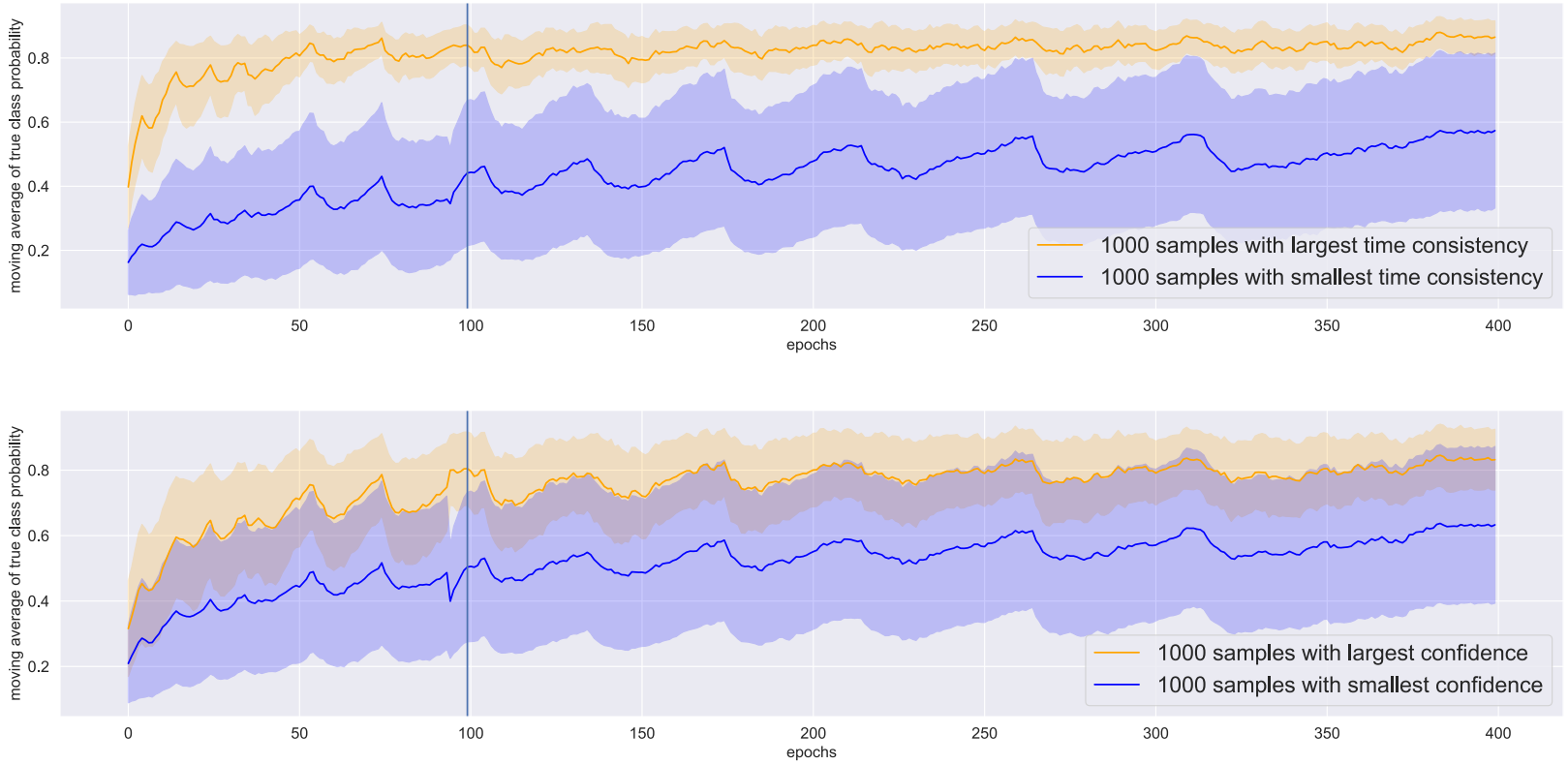
- Split CIFAR10 training set into two subsets of 15000 and 35000 samples.
- Train WideResNet-18-2 on the 15000 samples, test it on the 35000 samples.
- Time consistency performs better than confidence in identifying the unlabeled samples correctly predicted by the current model.



Computed time-consistency and confidence at epoch 100 of training WideResNet-28-2. The x-axis shows the validation samples selected using different thresholds on the two metrics (normalized to [0, 100]). The y-axis reports correct v.s. incorrect predictions over the selected samples.

Persistence of Time Consistency

- Time consistency performs better in predicting the future dynamics, i.e., it identifies samples whose predictions stay correct stably in the future.



- Computed time-consistency (top) and confidence (bot- tom) at epoch 100 of training WideResNet-28-2 on CIFAR10.
- Select the top 1000 and bottom 1000 validation samples based on the two metrics.
- Compare the moving average of true class probability of the selected samples across epochs.

TC-SSL Algorithm

- In each step, select unlabeled samples with large time-consistency and optimize our SSL objective on them.

- Add warm-start epochs and apply exponential weighted sampling to encourage exploration in early stages.

- Remove samples with extremely high confidence since they contribute nearly zero gradients.

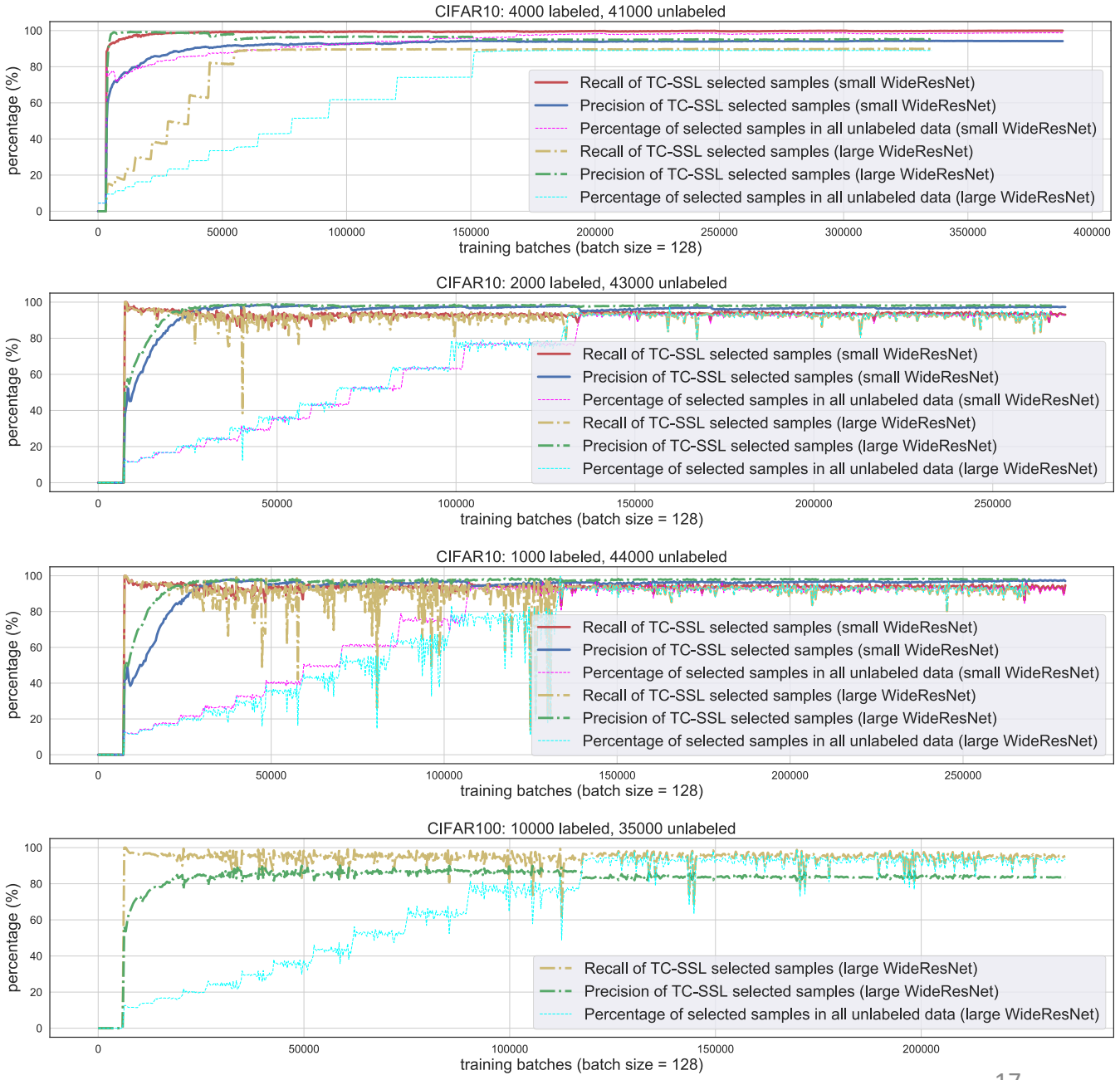
- Follow previous works: Mix-Up, sharpen predicted probability as pseudo target, duplicate labeled data to similar amount of selected unlabeled data, etc.

Algorithm 1 Time-Consistent SSL (TC-SSL)

input: $\mathcal{U}, \mathcal{L}, \pi(\cdot;\eta), \eta^{1:T}, f(\cdot;\theta), G(\cdot)$;
hyperparameters: $T_0, T, \lambda_{cs}, \lambda_{ct}, \lambda_{ce}, \gamma_\theta, \gamma_c, \gamma_k$;
initialize: θ^0, k^1;
for $t \in \{1, \cdots, T\}$ **do**
  if $t \leq T_0$ **then**
    $\theta^t \leftarrow \theta^{t-1} + \pi\left(\sum_{(x,y)\in\mathcal{L}} \nabla_\theta \ell_{ce}(x, y; \theta^{t-1}); \eta^t\right)$
  else
    $S^t = \text{argmax}_{S:S\subseteq\mathcal{U},|S|=k^t} \sum_{x\in S} c^t(x)$ or
    Draw k^t samples from $\Pr(x \in S^t) \propto \exp(c^t(x))$;
    $\theta^t \leftarrow \theta^{t-1} + \pi\left(\nabla_\theta L^t(\theta^{t-1}); \eta^t\right)$ (ref. Eq. (11));
  end if
  $p^t(x) \leftarrow \frac{\exp(f(x;\theta^t)[y])}{\sum_{y'=1}^{C} \exp(f(x;\theta^t)[y'])}$, $\forall y \in [C], x \in \mathcal{U}$;
  if $t = 1$ **then**
    $\overline{\theta^t} \leftarrow \theta^t, c^t(x) \leftarrow 0, \ \forall x \in \mathcal{U}$
  else
    Compute $a^t(x)$ (ref. Eq (1)), $\forall x \in \mathcal{U}$;
  end if
  $c^{t+1}(x) \leftarrow \gamma_c(-a^t(x)) + (1-\gamma_c)c^{t-1}(x), \ \forall x \in \mathcal{U}$;
  $\overline{\theta^{t+1}} \leftarrow \gamma_\theta \theta^t + (1-\gamma_\theta)\overline{\theta^t}$;
  $k^{t+1} \leftarrow (1+\gamma_k) \times k^t$;
end for

Quality of Selected Pseudo Targets in TC-SSL

- TC-SSL produces a curriculum of unlabeled data whose pseudo targets are of high precision and recall throughout the course of training;
- TC-SSL gradually increase the use of unlabeled data rather than adding all of them to training at the very beginning.



Experimental Results

- TC-SSL achieves SOTA performance on CIFAR10, CIFAR100, STL10 of different labeled/unlabeled splittings (more results in paper).

Table 1. Test error rate (mean±variance) of SSL methods training a small WideResNet and a large WideResNet on **CIFAR10**. Baselines: Pseudo Label (Lee, 2013), Π-model (Sajjadi et al., 2016), VAT (Miyato et al., 2019), Mean Teacher (Tarvainen & Valpola, 2017), MixMatch (Berthelot et al., 2019), ReMixMatch (Berthelot et al., 2020).

Benchmark	CIFAR10 (small WideResNet-28-2)				CIFAR10 (large WideResNet-28-135)			
labeled/unlabeled	500/44500	1000/44000	2000/43000	4000/41000	500/44500	1000/44000	2000/43000	4000/41000
Pseudo Label	40.55 ± 1.70	30.91 ± 1.73	21.96 ± 0.42	16.21 ± 0.11	-	-	-	-
Π-model	41.82 ± 1.52	31.53 ± 0.98	23.07 ± 0.66	5.70 ± 0.13	-	-	-	-
VAT	26.11 ± 1.52	18.68 ± 0.40	14.40 ± 0.15	11.05 ± 0.31	-	-	-	-
Mean Teacher	42.01 ± 5.86	17.32 ± 4.00	12.17 ± 0.22	10.36 ± 0.25	-	-	-	-
MixMatch	9.65 ± 0.94	7.75 ± 0.32	7.03 ± 0.15	6.24 ± 0.06	8.44 ± 1.04	7.38 ± 0.63	6.51 ± 0.48	5.12 ± 0.31
ReMixMatch	-	5.73 ± 0.16	-	5.14 ± 0.04	-	-	-	-
TC-SSL (ours)	9.14 ± 0.88	6.15 ± 0.23	5.85 ± 0.10	5.07 ± 0.05	6.04 ± 0.39	3.81 ± 0.19	3.79 ± 0.21	3.54 ± 0.06

Experimental Results

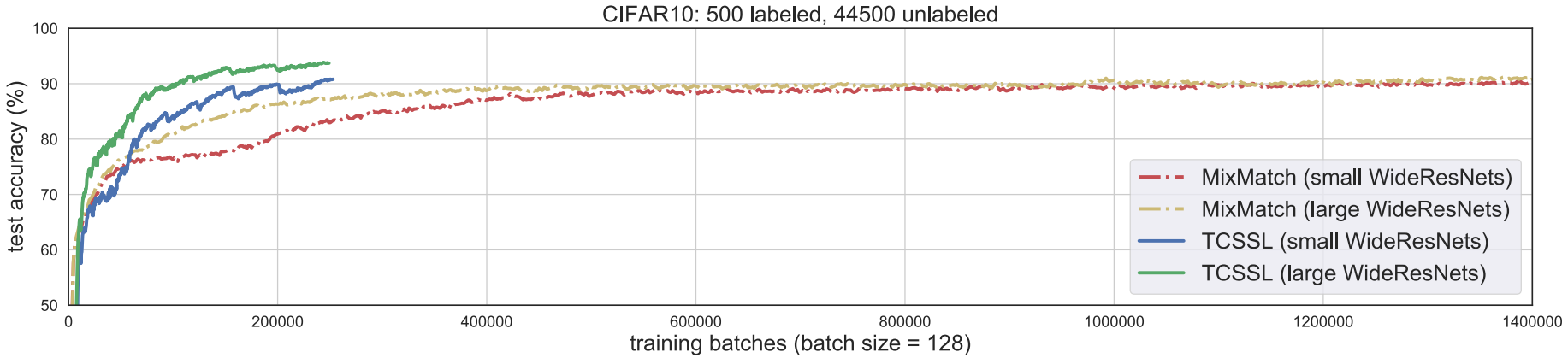
- TC-SSL achieves SOTA performance on CIFAR10, CIFAR100, STL10 of different labeled/unlabeled splittings (more results in paper).

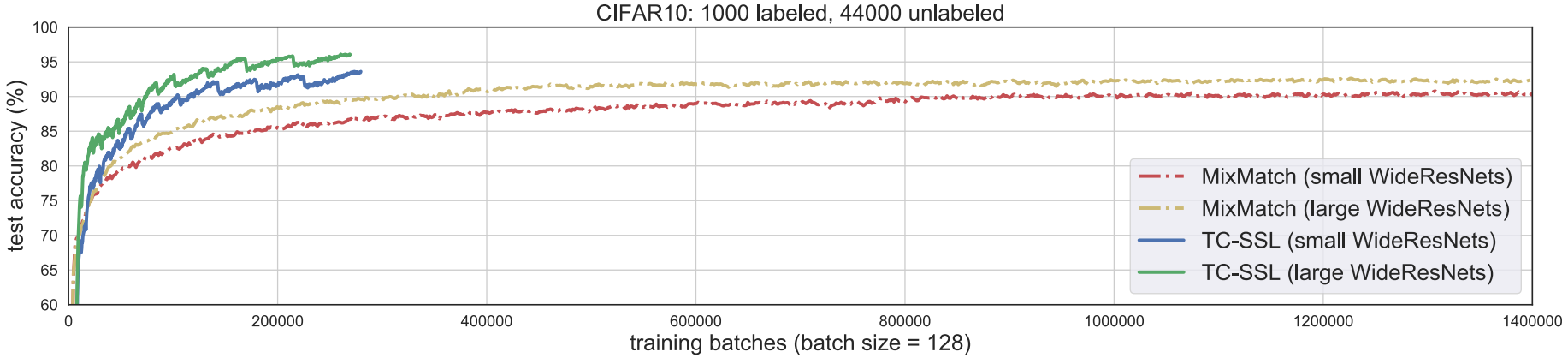
Benchmark	STL10
labeled/unlabeled	5k/100k
CutOut (11.0M)	12.74
IIC (N/A)	11.20
MixMatch (23.8M)	5.59
TC-SSL (ours, 5.9M)	4.82

Benchmark	CIFAR100 (WideResNet-28-135)		
labeled/unlabeled	2500/42500	5000/40000	10000/35000
SWA	-	-	28.80
MixMatch	44.20 ± 1.18	34.62 ± 0.63	25.88 ± 0.30
TC-SSL (ours)	31.95 ± 0.55	26.98 ± 0.51	22.10 ± 0.37

Experimental Results

- TC-SSL significantly improves SSL efficiency.
- It achieves high accuracy using much fewer but more informative and time-consistent training batches with more accurate pseudo targets.





Ablation Study

- Test error rate (mean±variance) of TC-SSL variants training WideResNet on CIFAR10;
- no consistency: TC-SSL without consistency loss;
- no contrastive: TC-SSL without contrastive loss;
- no PseudoLabel: TC-SSL without cross entropy loss for unlabeled data;
- no TC-selection: replace TC-based selection/sampling with uniform sampling.

labeled/unlabeled	500/44500	1000/44000	2000/43000	4000/41000
TC-SSL (ours)	6.04 ± 0.39	3.81 ± 0.19	3.79 ± 0.21	3.54 ± 0.06
TC-SSL (no consistency)	7.51 ± 0.56	5.31 ± 0.23	3.82 ± 0.20	3.58 ± 0.06
TC-SSL (no contrastive)	7.56 ± 0.52	5.35 ± 0.28	3.96 ± 0.25	3.66 ± 0.08
TC-SSL (no PseudoLabel)	41.05 ± 2.32	23.64 ± 1.17	14.37 ± 0.69	9.87 ± 0.22
TC-SSL (no TC-selection)	12.25 ± 0.81	6.39 ± 0.44	4.68 ± 0.35	4.05 ± 0.13

Take-home Messages

Time-consistency is critical to semi-supervised learning;

We derive a novel time-consistency metric with theoretical support on avoiding catastrophic forgetting and plenty of empirical evidences;

We provide a recipe of self-supervision losses: consistency + contrastive;

TC-SSL, the proposed algorithm, achieves SOTA performance on several SSL benchmarks and considerably improves efficiency.

Thank you!

- For questions and discussions, please join our Q&A session.
  - July 15 Web 10:00 AM PDT
  - July 15 Web 23:00 PM PDT