Semi-Supervised Learning with Scarce Annotations

Sylvestre-Alvise Rebuffi* Sebastien Ehrhardt* Kai Han*
Andrea Vedaldi Andrew Zisserman
Visual Geometry Group, University of Oxford
{srebuffi,hyenal,khan,vedaldi,az}@robots.ox.ac.uk

QikuiZhu 08/22/2019

Rebuffi, Sylvestre-Alvise, et al. "Semi-Supervised Learning with Scarce Annotations." *arXiv preprint arXiv:1905.08845* (2019).

Motivation

1. Data collection is increasingly inexpensive, but data annotation still involves manual and thus expensive labor.

2. Semi-supervised learning (SSL) can significantly reduce the cost of learning new models by using large datasets of which only a small proportion comes with manual labels.

3. In fact, it is difficult to learn effectively from a label-deficient dataset with a small number of data with known labels and a very large number of data points with unknown labels.

4. When the ratio of labelled and unlabeled data is very unbalanced, the most likely result is that the neural network would overfit the few labelled data, which would then cease to have an influence on the unlabeled data.

Contribution

1. They propose to use self-supervision to bootstrap a good representation of network. They exploit the ability of deep network to transfer effectively between tasks and re-use the pre-trained feature to initialize SSL.

2. A new SSL trained algorithm be proposed. Instead of fitting labelled and unlabeled data simultaneously, they alternate between two steps of optimization. The information flow between the two steps is carefully controlled to minimize the risk of overfitting.

RotNet (Self-supervision)

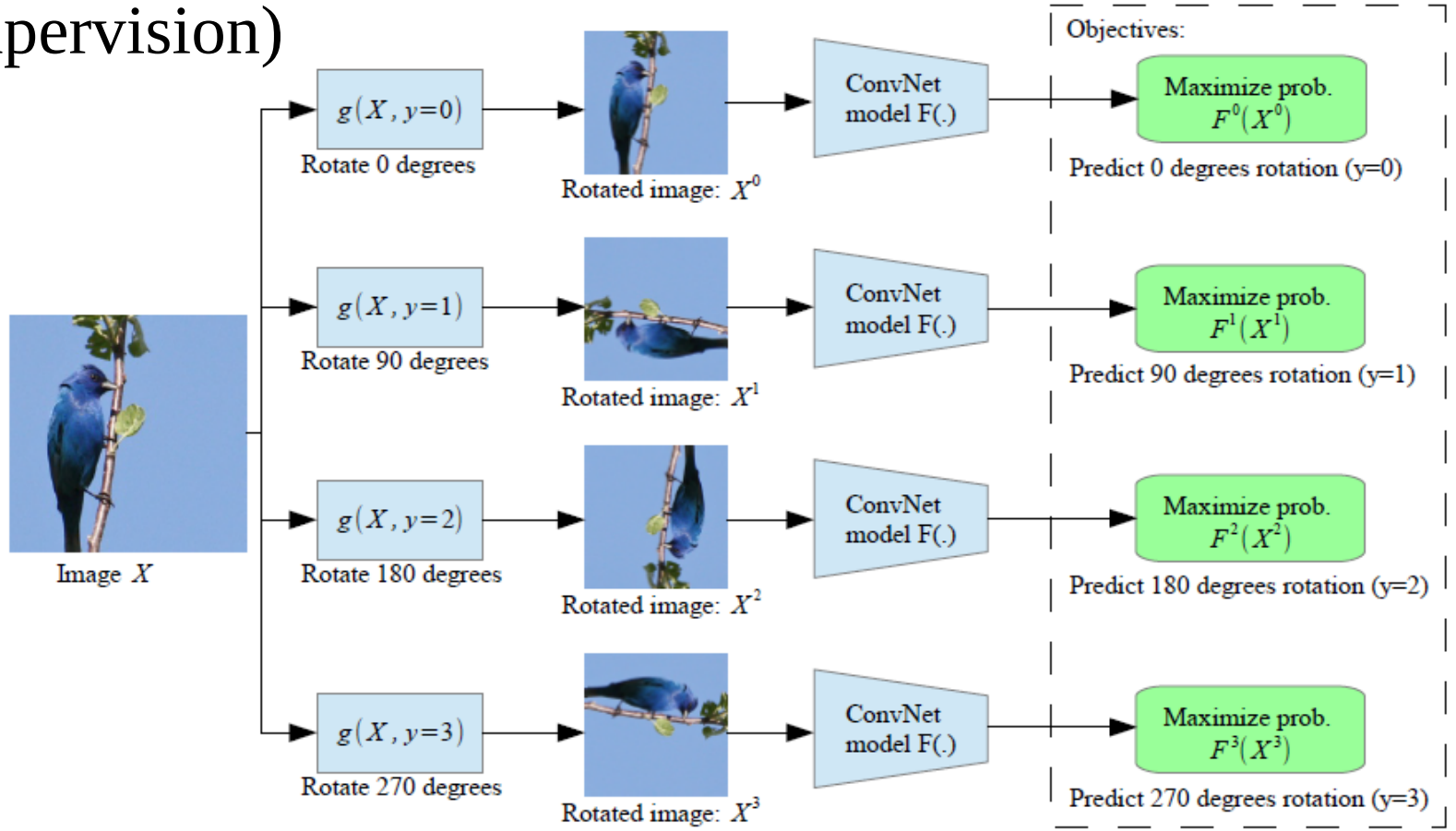

Figure 2: Illustration of the self-supervised task that we propose for semantic feature learning. Given four possible geometric transformations, the 0, 90, 180, and 270 degrees rotations, we train a ConvNet model $F(.)$ to recognize the rotation that is applied to the image that it gets as input. $F^y(X^{y^*})$ is the probability of rotation transformation y predicted by model $F(.)$ when it gets as input an image that has been transformed by the rotation transformation y^*.

S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In ICLR, 2018.

Method

Algorithm 1 Proposed Alternative Optimisation Algorithm

Preparation phase:
Train a self-supervised method on the whole dataset and freeze the first blocks' weights. Discard the last layer and replace with suitably dimensioned one. The trainable weights now form a network N_t
Main Loop:
for $i \in \{1, \ldots, N\}$ **do**
    Supervised-training: Fine-tune N_t classification layer on the labelled subset with cross-entropy loss $\mathcal{L}_{labelled}$.
    Labels assignement: Use N_t to assign a label y_i to each unlabelled sample x_i.
    Dataset split: Create a training set T_i from a random split of the unlabelled data.
    Restart: Reassign N_t to the weights extracted from the preparation phase.
    Unsupervised-training: Train N_t on the unlabelled metaset T_i with consistency and pseudo labelling loss: $\mathcal{L}_{unlabelled} = 0.5\mathcal{L}_{temp} + 0.5\mathcal{L}_{pseudo}$.
end for

Phase one: fitting the labelled data.

We optimize the cross-entropy loss on the labelled set and train the model for a few epochs. In this part, only fine-tune the final classification layer of the network.
Then, the classifier is used to generate pseudo-labels for unlabeled data.

Phase two: fitting the unlabeled data.

Reset the model with the parameters of network by the weight learned by self - supervised learning;
Then, fine-tune the whole architecture on the unlabeled set using a loss that is a weighted average of a term fitting the pseudo-labels estimated in the first phase, and a second temporal consistency term

$$L = L_{pseudo} + L_{temp}$$

L_{pseudo} :cross-entropy loss

L_{temp} : KL-divergence between p_i^{t-1} and p_i^t

$$D_{\mathrm{KL}}(P \parallel Q) = -\sum_{x \in \mathcal{X}} P(x) \log\left(\frac{Q(x)}{P(x)}\right)$$

Alternate optimization

During training, current state-of-the-art algorithms generally use mini-batches containing a mix of labelled and unlabeled data. They use a sum of two losses corresponding to two different tasks: the first one enforces the classification of the labelled samples with a high confidence while the second one enforces prediction consistency across the unlabeled samples.

Drawback

Since both losses are trained jointly, they must be balanced with hyper parameter tuning. This is particularly true when using very few label instances as there is a large information imbalance between the two losses.

Furthermore, since mini batches are generally divided between labelled and unlabeled data, the same samples from the annotated set will be seen very frequently, thus overfitting these samples.

Method

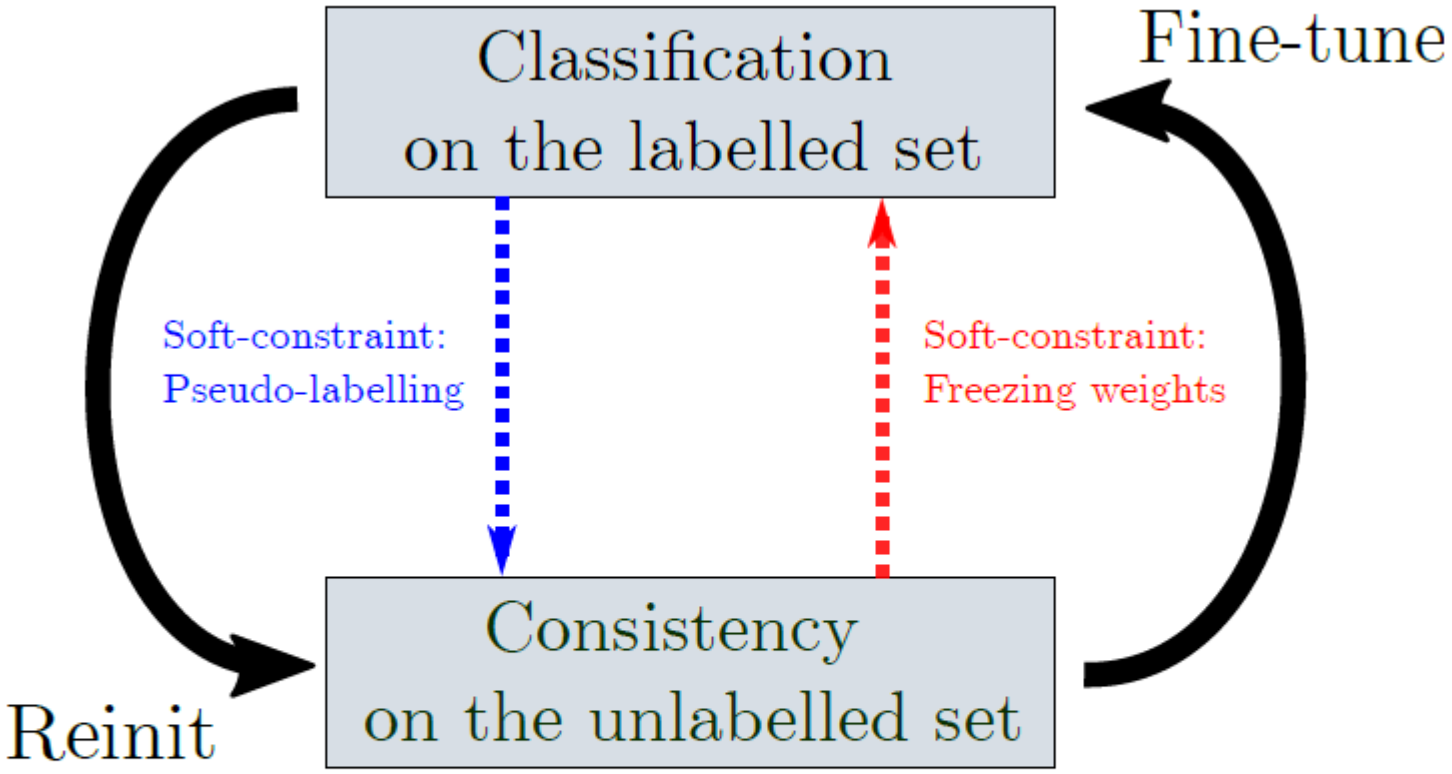

Figure 2. **Overview of our alternating optimisation method.** Starting from a pre-trained network with a self-supervised learning method we train successively on the labelled and unlabelled set. Every time a soft constraint is enforced from previous optimisation.

Phase one: fitting the labelled data.

Phase two: fitting the unlabeled data.

Experiments

Datasets

SVHN: A Street View House Numbers dataset including 10 classes (0-9) of colored digit images from Google Street View. The classification task is to recognize the central digit of each image. We use the format-2 version that provides cropped images sized at 32 x 32, and the standard 73,257/26,032 training/test data split.

CIFAR-10: A natural images dataset containing 50,000 training and 10,000 test image samples from 10 object classes. Images have a 32 x 32 resolution and are evenly divided among classes.

CIFAR-100: A dataset (with the same image size as CIFAR-10) containing 50,000/10,000 training/test images from 100 more fine-grained classes with subtle inter-class visual discrepancy.

Implementation details: RotNet as the self supervision method, ResNet-18 is the classification model.

Impact of self-supervision

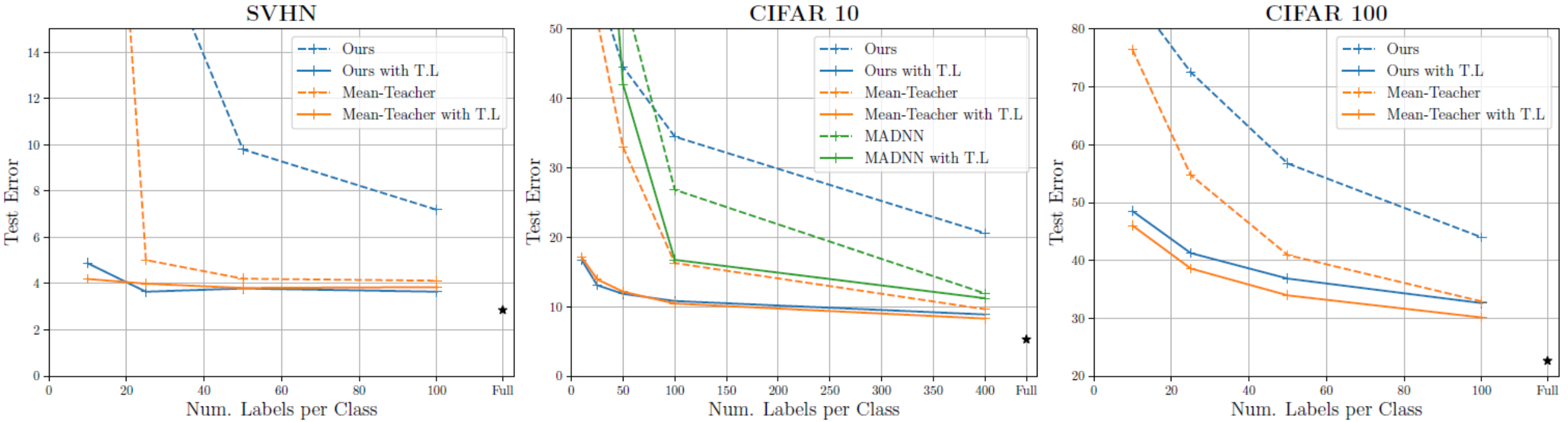

Figure 3. **Impact of Tranfer learning for SSL classification benchmark.** For each dataset we vary the number of labelled data per class in 10, 25, 50, 100 with an additional experiment on CIFAR-10 with 400 labels per class. We note that for every method, models benefit from self-supervision. The 'with T.L' indicates a model pretrained with RotNet[7]. In addition we can see that our method is getting competitive results with Mean Teacher. The star point '*' denotes our network trained with supervision only on the full training set. The shown results are the average of 10 runs per setting.

Ablation study

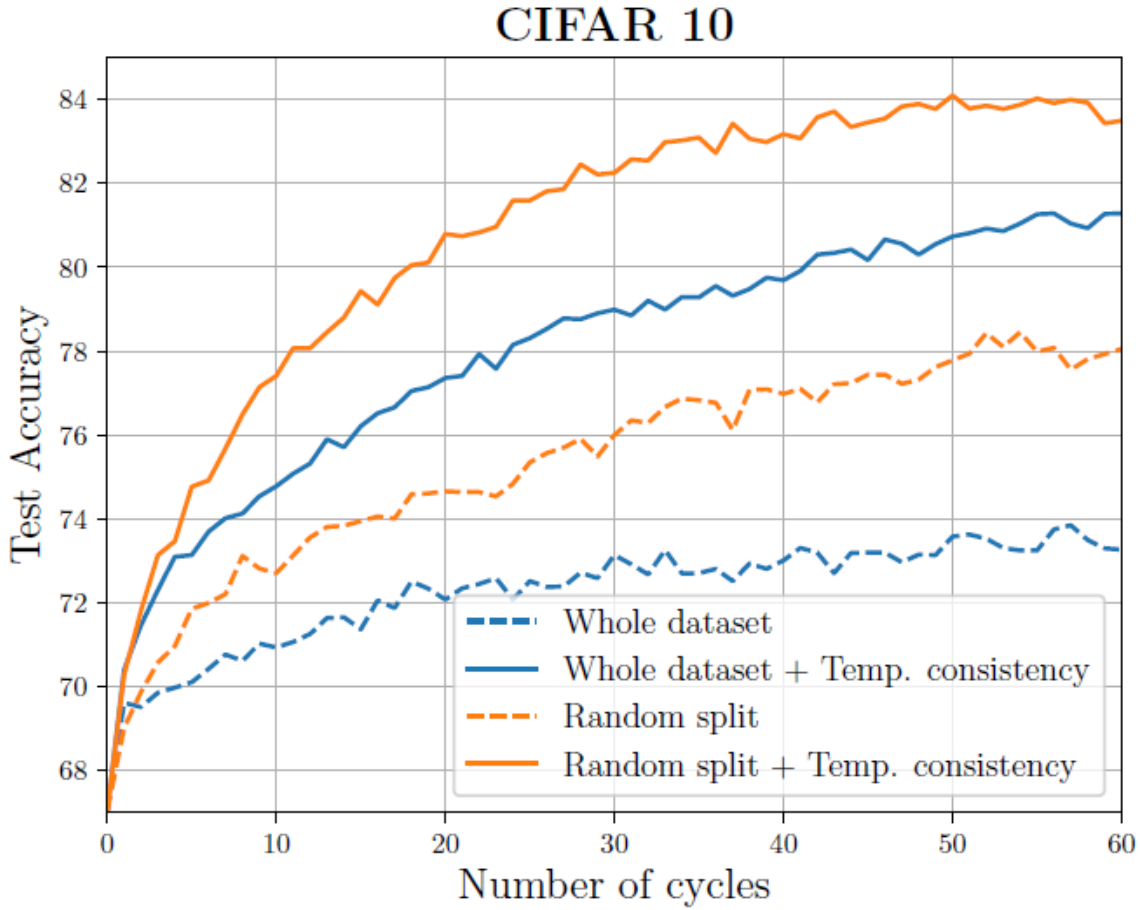

Figure 4. **Ablation study.** Starting from a pre-trained model, our alternating optimisation method makes steady progress over cycles. Dataset random splitting is also effectively speeding-up training. Default case uses $\mathcal{L}_{unlabelled} = \mathcal{L}_{pseudo}$. 'Temp. consistency' indicates that we use both $\mathcal{L}_{temp}$ and $\mathcal{L}_{pseudo}$.

Evaluate the random splitting of the unlabeled set through training cycles and temporal consistency term in the loss function.

A refining SSL algorithm

	CIFAR-10	CIFAR-100	
Labels per class	10	10	25
MT	32.4	23.6	45.7
MT + Self-Sup	86.4	53.7	60.7
MT + Self-Sup + Refinement	**87.7**	**58.3**	**63.9**

Table 1. **Our method applied as a refinement of Mean Teacher ('MT').** For different numbers of labels per class our method always improve the testing accuracy by a big margin.

Transfer learning from different tasks

	C-10	C-100	P-10	MIT-67
Fully Sup.	96.6	82.5	83.3	75.5
Labelled Set	82.5±1.0	62.8±0.6	66.4±1.7	55.1±0.8
M.T.[33]	88.0±1.5	63.3±0.7	72.8±1.2	58.8±1.3
Ours	**93.5±0.4**	**69.2±0.4**	**75.8±1.7**	**63.0±0.7**

Table 2. **Transfer Learning: Test accuracy on the target task.** Transfer is from training on ImageNet. We compare our method with Mean Teacher (M.T.) and supervision on the labelled set only ('Labelled set') for 10 labelled sample per class. Our method consistently outperforms both method on this task and achieves results nearly as good as the one obtained by training on the full labelled set.

Mean Teacher

consistency cost

$$J(\theta) = \mathbb{E}_{x,\eta',\eta}\left[\|f(x,\theta',\eta') - f(x,\theta,\eta)\|^2\right]$$

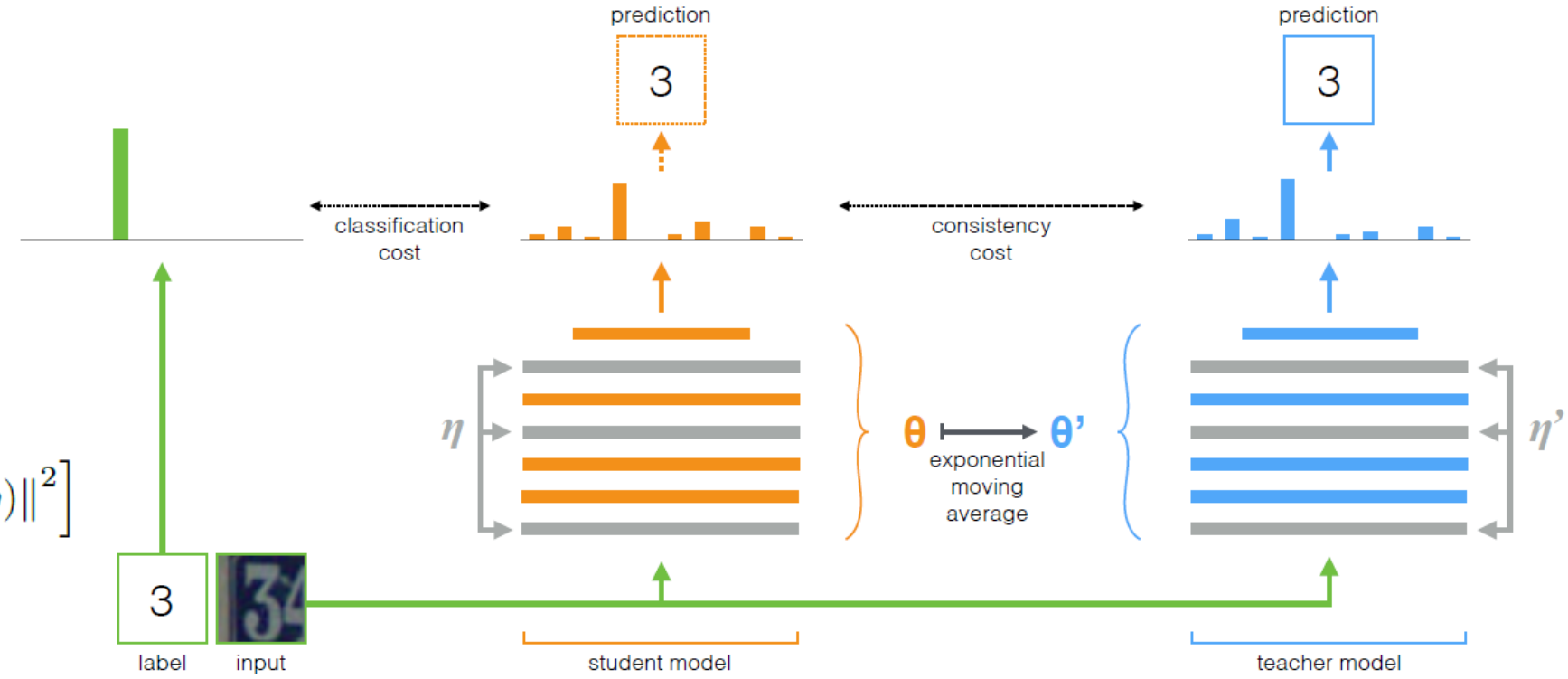

Figure 2: The Mean Teacher method. The figure depicts a training batch with a single labeled example. Both the student and the teacher model evaluate the input applying noise (η, η') within their computation. The softmax output of the student model is compared with the one-hot label using classification cost and with the teacher output using consistency cost. After the weights of the student model have been updated with gradient descent, the teacher model weights are updated as an exponential moving average of the student weights. Both model outputs can be used for prediction, but at the end of the training the teacher prediction is more likely to be correct. A training step with an unlabeled example would be similar, except no classification cost would be applied.

A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi supervised deep learning results. In NeurIPS, 2017.