VCC-INFUSE: Towards Accurate and Efficient Selection of Unlabeled Examples in Semi-supervised Learning

Shijie Fang[1,2] , **Qianhan Feng**[1] , **Tong Lin**[1]
[1]National Key Laboratory of General Artificial Intelligence,
School of Intelligence Science and Technology, Peking University
[2]Google, Shanghai, China
shijiefang@google.com, fengqianhan@stu.pku.edu.cn, lintong@pku.edu.cn

Abstract

Despite the progress of Semi-supervised Learning (SSL), existing methods fail to utilize unlabeled data effectively and efficiently. Many pseudo-label-based methods select unlabeled examples based on inaccurate confidence scores from the classifier. Most prior work also uses all available unlabeled data without pruning, making it difficult to handle large amounts of unlabeled data. To address these issues, we propose two methods: Variational Confidence Calibration (**VCC**) and Influence-Function-based Unlabeled Sample Elimination (**INFUSE**). VCC is a universal plugin for SSL confidence calibration, using a variational autoencoder to select more accurate pseudo labels based on three types of consistency scores. INFUSE is a data pruning method that constructs a core dataset of unlabeled examples under SSL. Our methods are effective in multiple datasets and settings, reducing classification error rates and saving training time. Together, VCC-INFUSE reduces the error rate of FlexMatch on the CIFAR-100 dataset by 1.08% while saving nearly half of the training time.

1 Introduction

Deep neural networks underpin various machine learning applications, with their success attributed in part to extensive labeled datasets like ImageNet [Deng *et al.*, 2009] and COCO [Lin *et al.*, 2014]. However, the process of collecting and annotating large datasets is resource-intensive and raises privacy concerns, making the acquisition of unlabeled data a more feasible and cost-effective alternative.

To address the challenge of limited labeled examples, semi-supervised learning (SSL) has gained prominence for leveraging abundant unlabeled data. Pseudo-labeling is a common SSL approach, as demonstrated by FixMatch [Sohn *et al.*, 2020]. FixMatch generates pseudo labels for unlabeled data based on model predictions. The threshold-based selection module in FixMatch filters examples with confidence scores surpassing a fixed threshold τ for training. Formally, the loss on unlabeled data is defined as:

$$\mathcal{L}_{unlab} = \sum_i \mathbb{1}(\max(c_i) \geq \tau)\mathcal{L}(\hat{c}_i, \widetilde{c}_i), \tag{1}$$

where $\mathcal{L}(\hat{c}_i, \widetilde{c}_i)$ represents the loss between class label and confidence distribution.

Despite FixMatch's wide adoption, it may encounter challenges in utilizing unlabeled examples effectively and efficiently. Specifically, (1) **Incorrect pseudo labels** may arise due to calibration errors in model predictions, leading to unreliable performance. (2) The **significant computation cost** involved in forwarding the entire unlabeled dataset for pseudo label selection may be alleviated by dynamically pruning the dataset. This ensures that only informative data points contribute to the model's decision boundary, reducing computation overhead and accelerating convergence.

In this paper, we propose solutions to address these challenges, enhancing the reliability and efficiency of SSL methods based on pseudo-labeling. To address the first issue, we introduce Variational Confidence Calibration (VCC), a method aimed at obtaining calibrated confidence scores for pseudo label selection. The calibrated confidence score, closely aligned with the ground-truth probability of correct predictions, serves as a more reliable metric for choosing pseudo-labeled examples. While confidence calibration is a well-explored concept in fully-supervised settings, its application in semi-supervised learning (SSL) is more challenging due to the absence of ground-truth labels. To overcome this challenge, we utilize three consistency scores to assess prediction stability. By simultaneously considering both stability and confidence, we approximate calibrated confidence scores. Additionally, a variational autoencoder enhances stability by reconstructing the calibrated confidences.

To address the second issue, we propose INFUSE (Influence Function-based Unlabeled Sample Elimination), a method leveraging influence functions [Koh and Liang, 2017] to compute the importance of each unlabeled example. INFUSE dynamically retains data points with the highest importance, forming a smaller core set to replace the entire dataset. This core set allows for faster model convergence, reducing computation costs during training. The combined VCC-INFUSE method enhances prediction accuracy while minimizing training costs.

In summary, this paper makes following contributions:

- We propose the VCC method, which generates well-calibrated confidence scores for more accurate pseudo labels, enhancing model accuracy. As a flexible, plug-and-play module, VCC can be seamlessly integrated with existing SSL methods.
- We introduce the INFUSE method, which dynamically prunes unimportant unlabeled examples to expedite convergence and reduce computation costs during training.
- The effectiveness of our methods is demonstrated across multiple datasets and various settings.

2 Related Work

Semi-supervised Learning. FixMatch [Sohn *et al.*, 2020] stands out as one of the most widely adopted SSL methods. FixMatch utilizes a weakly-augmented unlabeled example to obtain a one-hot pseudo label, followed by training the model on strongly-augmented examples to produce predictions consistent with the pseudo label. FlexMatch [Zhang *et al.*, 2021] introduces an adaptive threshold strategy, tailored to different learning stages and categories. SimMatch [Zheng *et al.*, 2022] considers both semantic and instance similarity, promoting consistent predictions and similar similarity relationships for the same instance. Additionally, explicit consistency regularization is employed in various SSL methods [Laine and Aila, 2016; Berthelot *et al.*, 2020; Miyato *et al.*, 2019; Ganev and Aitchison, 2020; Chen *et al.*, 2023; Li *et al.*, 2021; Feng *et al.*, 2024; Lee *et al.*, 2021].

Confidence Calibration. [Guo *et al.*, 2017] identified the calibration problem in modern classifiers and proposed Temperature Scaling (TS) to rescale confidence distributions, preventing over-confidence. Ensemble TS [Zhang *et al.*, 2020] extends TS's representation ability by expanding the parameter space. Additionally, [Kumar *et al.*, 2018] introduces the MMCE method, a trainable calibration regularization based on Reproducing Kernel Hilbert Space (RKHS). Notably, these methods are designed for fully-supervised settings where ground-truth labels are available.

Core Set Selection. While most core set selection methods focus on the fully-supervised setting, our work aligns more closely with the semi-supervised learning context. [Paul *et al.*, 2021] proposes the EL2N method, measuring the importance of an example based on the norm of the loss. EL2N significantly reduces training time with a minimal impact on accuracy. GradMatch [Killamsetty *et al.*, 2021a] extends the core dataset to a weighted set using a submodular function. In the realm of SSL, RETRIEVE [Killamsetty *et al.*, 2021b] addresses core set selection as an optimization problem. However, RETRIEVE's optimization function only considers the loss labeled set, potentially deviating from the desired objective of minimizing loss on the validation set.

3 Confidence Calibration with VCC

Many existing calibration methods are ill-suited for SSL due to the absence of ground-truth labels for unlabeled examples. Directly using the original confidence score for pseudo label selection can yield noisy results. To tackle this challenge, we introduce three different consistency scores (s^{ens}, s^{tem}, and s^{view}) to simultaneously gauge the stability of predictions. By combining these three scores, we obtain the approximated calibrated confidence $\tilde{r}$, which is closer to the probability of an example being correctly classified. However, $\tilde{r}$ is not directly utilized for pseudo label selection, as the process of estimating $\tilde{r}$ from three consistency scores can still be unstable for some examples.

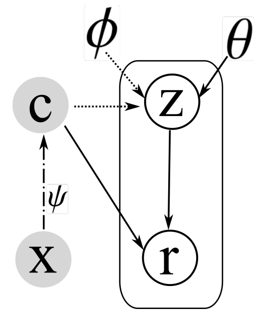

Figure 1: The graphical model of VCC. Here x is the input, c is the originally predicted confidence distribution, z is the latent variable sampled from the encoder, and r is the reconstructed confidence for pseudo label selection. Dash-dotted line denotes the original prediction function $p_\psi(c|x)$. Solid lines denote the generative model $p_\theta(r|c, z, x)$. Dashed lines denote the approximation $q_\phi(z|c, x)$ to the intractable posterior $p_\theta(z|c, r, x)$. The variational parameter ϕ is learned jointly with the generative model parameter θ.

To mitigate this instability, we introduce a Variational Autoencoder (VAE) to reconstruct $\tilde{r}$ for pseudo label selection. The graphical model and framework illustration of VCC are provided in Fig.1 and 2, respectively. The VAE is learned jointly with the original classifier during training, where $\tilde{r}$ serves as the "ground-truth" for calculating the reconstruction loss. For pseudo label selection, we leverage the output of the VAE as the calibrated confidence.

3.1 Ensemble Consistency

From a Bayesian perspective, the parameters θ of a model are sampled from a probability distribution over the training set D. Model's prediction for sample x can be formulated as:

$$p(y|x, D) = \int p(y|x, \theta)p(\theta|D)\mathrm{d}\theta, \tag{2}$$

where $p(y|x, \theta)$ represents the probability distribution of the label y of x given the parameters θ, and $p(\theta|D)$ represents the probability distribution of the model parameters θ trained on the dataset D. A single model may provide incorrect predictions, for example, x due to randomness and noise, even if the confidence is high. Considering the entire parameter space, if all model parameters yield consistent predictions for x, the result is more convincing. In this case, the prediction can be viewed as an ensemble of predictions from multiple models.

However, due to the large parameter space of θ, direct computation of Equation 2 is intractable. Therefore, we apply Monte-Carlo Dropout [Gal and Ghahramani, 2016] on the linear layer to approximate the computation of Equation 2. The feature map is cloned by K copies, followed by a Dropout layer to randomly eliminate neural connections in the classification head to obtain predictions. By doing so, the model would generate K estimated confidence distributions of example i, and the expectation can be treated as the ensemble of K different models:

$$\hat{\mathbf{y}}_i = p(y|x, \text{Dropout}(\theta)), \quad \tilde{\mathbf{y}} = \frac{1}{K}\sum_{i=1}^{K}\hat{\mathbf{y}}_i. \tag{3}$$

Then, entropy is employed as the ensemble-consistency score to measure the different models' consistency of example: $s^{ens} = -\sum_{c=1}^{M}\tilde{\mathbf{y}}_c \log \tilde{\mathbf{y}}_c$.

3.2 Temporal Consistency

In SSL, parameters are updated frequently during training, making the decision boundaries change all the time. Some examples may shift from one side of the decision boundary to the other side after parameter updates, bringing a change in classification results. In this case, the prediction results of many examples may be rather unstable. If these examples are used in training, it may result in incorrect pseudo labels and hinder the model's performance.

To measure the stability of prediction results between different steps, we propose the temporal consistency score, which considers the changes in the confidence distribution of an example between different epochs. Specifically, let y^t represent the confidence distribution of an example at epoch t. The temporal consistency score can be calculated as:

$$\begin{aligned} s^{tem} &= D_{KL}\left(y^t \middle\| \frac{1}{K}\sum_{k=1}^{K} y^{t-k}\right) \\ &= \sum_{c=1}^{M} y_c^t \log\left(\frac{y_c^t}{\frac{1}{K}\sum_{k=1}^{K} y_c^{t-k}}\right), \end{aligned} \tag{4}$$

where D_{KL} represents the Kullback-Leibler Divergence, M is the number of classes, K represents the window size. In experiments, we empirically set $K = 1$ to preserve the sensitivity of abnormal confidences. Although both consider the problem from the perspective of time, our method differs a lot from the method proposed by [Zhou *et al.*, 2020].

3.3 View Consistency

Multi-view learning [Xu *et al.*, 2015] aims to leverage multiple perspectives to predict data, allowing different predictors to correct predictions collectively.

In semi-supervised learning (SSL), obtaining models with different views often involves dividing the entire dataset into multiple subsets for training multiple models. However, this incurs high model training costs, and the volume of labeled data in each subset may be too small to train a decent model. To address this, we use Exponential Moving Average (EMA) to construct models with different views. The original model parameter θ is updated using gradient descent, while θ_{ema} is updated using the EMA scheme:

$$\theta_{ema}^t = \theta^t \cdot \beta + \theta_{ema}^{t-1} \cdot (1-\beta), \tag{5}$$

where β is a decay hyperparameter. These can be treated as two different views from the same network structure.

A typical classification model consists of a feature extraction network (backbone) and a classification head (linear layer). To increase the difference between two views, we adopt a cross-feature trick. The backbone of each view first extracts features from input, which are then fed into the classification head of another view. This can be formulated as:

$$y = p(y|x, \theta^{backbone}, \theta_{ema}^{head}), \tag{6}$$

$$y_{ema} = p(y|x, \theta_{ema}^{backbone}, \theta^{head}). \tag{7}$$

After obtaining the outputs, the Kullback-Leibler (KL) divergence is used to measure the consistency between them:

$$s^{view} = D_{KL}\left(y || y_{ema}\right). \tag{8}$$

It may seem like temporal consistency and view consistency overlap to some extent, as the predictions of the EMA model used in view consistency can also be considered an ensemble of predictions from past epochs. The difference is that the cross-feature trick is used in the computation of view consistency, which enforces this metric to focus more on consistency over multiple views rather than multiple time steps.

3.4 Approximation of Calibrated Confidence

We have introduced three scores to evaluate the stability of predictions. However, s^{ens}, s^{tem}, and s^{view} cannot be directly used for pseudo label selection, which is based on confidence scores. To address this, we propose a simple method to approximate calibrated confidence with the three consistency scores.

The consistency scores are first normalized and summed up as the stability score. First, a fixed-length queue q is maintained to record the historical predictions of the unlabeled samples in mini-batches. Since s^{ens}, s^{tem}, and s^{view} have different distributions, we normalize them with max-min normalization. Let u be the unlabeled example, the normalization is done as follows:

$$\tilde{s}_u^t = \frac{s_u^t - \min_{u' \in q}\left(s_{u'}^t\right)}{\max_{u' \in q}\left(s_{u'}^t\right) - \min_{u' \in q}\left(s_{u'}^t\right)}, \tag{9}$$

where $t = \{ens, tem, view\}$. After normalization, $\tilde{s}_u^{ens}$, $\tilde{s}_u^{tem}$, and $\tilde{s}_u^{view}$ are all real numbers ranging from 0 to 1. These consistency scores evaluate the stability of the examples. However, some hard-to-learn examples may also have stable predictions but low confidence scores. Hence, the three consistency scores are not enough to describe the reliability of the prediction. To address this, the original confidence score of the sample $\tilde{s}_u^{conf} = \max\left(y_u\right)$ is also used. Thus, an unlabeled sample u can be represented by a quadruple $(\tilde{s}_u^{ens}, \tilde{s}_u^{tem}, \tilde{s}_u^{view}, \tilde{s}_u^{conf})$.

The next problem is how to combine these four scores together for estimation. To avoid complex parameter tuning, VCC adopts a simple yet effective approach: taking the sum of their squares:

$$s_u = \left(\tilde{s}_u^{ens}\right)^2 + \left(\tilde{s}_u^{tem}\right)^2 + \left(\tilde{s}_u^{view}\right)^2 + \left(\tilde{s}_u^{conf}\right)^2. \tag{10}$$

According to the results in [Guo *et al.*, 2017], calibration errors mainly occur in the middle range of confidences, while samples with extremely low or high confidences tend to have

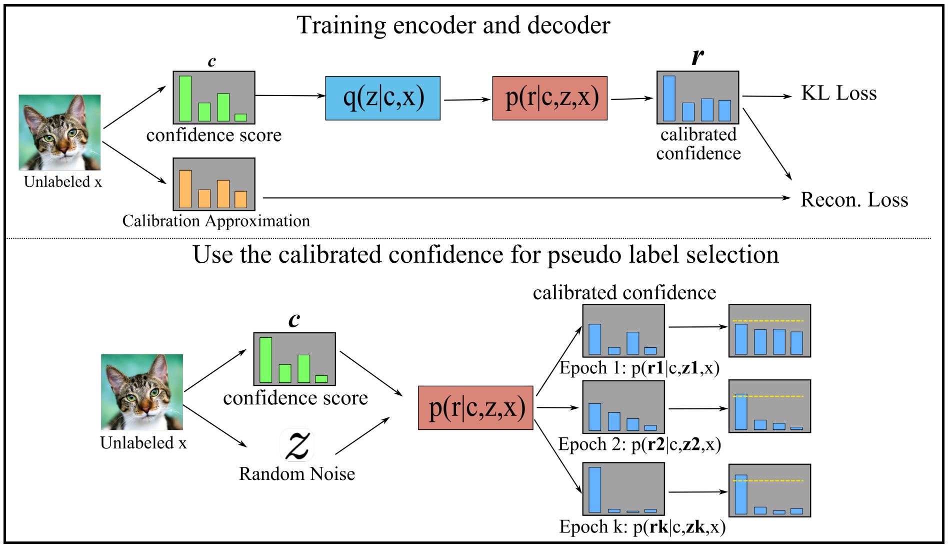

Figure 2: Illustration of VCC: training VAE and using the reconstructed confidence for pseudo label selection.

smaller calibration errors. Therefore, we approximately treat the lowest/highest confidence score in q as well-calibrated and employ interpolation to calculate the calibrated confidence scores for other examples. To further eliminate the unfairness between different categories, the interpolation operation only considers examples with the same pseudo labels as the current example u.

$$\begin{aligned}
&q' = \{e \mid e \in q, \arg\max \tilde{s}_e^{conf} = \arg\max \tilde{s}_u^{conf}\}, \\
&max_score = \max_{u' \in q'} (s_{u'}), min_score = \min_{u' \in q'} (s_{u'}), \\
&max_conf = \max_{u' \in q'} \left(\tilde{s}_{u'}^{conf}\right), min_conf = \min_{u' \in q'} \left(\tilde{s}_{u'}^{conf}\right),
\end{aligned} \tag{11}$$

and $\tilde{r}_u$ can be formulated as:

$$\tilde{r}_u = \frac{max\_score - s_u}{max\_score - min\_score} \cdot (max\_conf - min\_conf) + min\_conf.$$

3.5 Reconstruct $\tilde{r}_u$ with Variational Autoencoder

In Section 3.4, we combined three consistency scores to obtain $\tilde{r}_u$, which is the approximation of calibrated confidence scores. However, it may face instability due to the update of queue q and abnormal interpolation endpoints. To address this, we reconstruct the statistical-based $\tilde{r}_u$ in a learning-based way. Specifically, a Variational Autoencoder (VAE) is employed to generate the calibrated confidence score r_u for pseudo label selection, and $\tilde{r}_u$ is used as input for training the VAE.

We assume r is generated by the following random process, which includes two steps: (1) a hidden variable z sampled from a prior distribution $p_\theta(z)$; (2) a value r generated from the conditional distribution $p_\theta(r|c, z, x)$:

$$p_\theta(r|c, x) = \int_z p_\theta(z) p_\theta(r|z, c, x) \mathrm{d}z. \tag{12}$$

However, the marginal likelihood $p_\theta(r|c, x)$ is generally intractable. Hence another distribution $q_\phi(z|c, x)$ is introduced as the approximation of $p_\theta(z)$ (Please refer to Appendix[1] A for details):

$$\begin{aligned}
&\log p_\theta(r|c, x) = \int_z q_\phi(z|c, x) \log p_\theta(r|c, x) \mathrm{d}z \\
&\geq \mathbb{E}_{q_\phi(z|c,x)} \log p_\theta(r|c, z, x) - D_{KL}(q_\phi(z|c, x) \| p_\theta(z|c, x)).
\end{aligned} \tag{13}$$

The first term is the likelihood of calibration reconstruction (denoted as $\mathcal{L}_{VCC}^{recon}$), where $q_\phi(z|c, x)$ is the encoder to infer the hidden variable z, and $p_\theta(r|c, z, x)$ is the decoder to recover a calibrated confidence r. To compute the reconstruction loss, the approximated $\tilde{r}$ is used as the ground truth. Besides, z needs to be sampled from $q_\phi(z|c, x)$. Reparameterization trick [Kingma and Welling, 2014] is used to predict the mean and standard deviation of z. By setting $\epsilon \sim \mathcal{N}(0, 1)$, the reparameterization is formulated as $z = \mu(c, x) + \epsilon \cdot \sigma(c, x)$. For the second term, under the Gaussian assumptions of the prior $p_\theta(z|c, x) \sim \mathcal{N}(0, 1)$ and the approximator $q_\phi(z|c, x) \sim \mathcal{N}(\mu(c, x), \sigma^2(c, x))$, we have:

$$\begin{aligned}
\mathcal{L}_{VCC}^{KL} &\doteq D_{KL}(q_\phi(z|c, x) \| p_\theta(z|c, x)) \\
&= -\log \sigma + \frac{\mu^2 + \sigma^2}{2} - \frac{1}{2}.
\end{aligned} \tag{14}$$

The overall objective function can be formulated as:

$$\mathcal{L} = \mathcal{L}_{lab} + \lambda_{unlab} \cdot \mathcal{L}_{unlab} + \lambda_{VCC} \cdot \left(\mathcal{L}_{VCC}^{recon} - \mathcal{L}_{VCC}^{KL}\right). \tag{15}$$

Although a more accurate confidence score is generated by combining three consistencies, it is still not as optimal as the inaccessible ground-truth. This is because there are many other "nuisance" and untraceable factors that affect the pseudo label's approach toward the ground-truth, such as the

[1] Appendix is available at https://arxiv.org/abs/2404.11947.

randomness of the neural networks. Under these circumstances, directly approaching the unreliable target may still degrade performance. The original VAE is proposed to learn continuous distribution features from discontinuous distributions by sampling a hidden variable. This process is suitable for suboptimal pseudo label learning because the approach of the prediction to the generated pseudo label can be viewed as the process of the prediction approaching the ground-truth. Since eliminating those nuisance factors cannot be tractable, we use VAE to simulate this process instead of the MLP.

4 Core Set Selection with INFUSE

In the previous section, we introduced VCC framework, which ensures well-calibrated confidence scores to improve accuracy in pseudo label selection. Nonetheless, as discussed earlier, training the SSL model still encounters substantial computational expenses. Furthermore, the incorporation of additional encoder and decoder of VCC introduces an extra computation overhead. To address these challenges, we present INFUSE—a core set selection methodology aimed at efficient example selection. Based on the influence function [Koh and Liang, 2017], INFUSE allows for training the SSL model using only a subset of the complete unlabeled dataset, so that training time can be significantly reduced.

In SSL, the model should minimize the loss on the validation set to obtain the highest generalization accuracy:

$$\min \mathcal{L}(V, \theta^*), \quad \text{s.t. } \theta^* = \arg\min_{\theta} \; R(\theta), \tag{16}$$

$$\begin{aligned} R(\theta) \doteq & \mathbb{E}_{(x,y)\in S}\left[H(q_x, y)\right] \\ & + \lambda \cdot \mathbb{E}_{u\in U}\left[\mathbb{1}\left(\max(q_u) \geq \tau\right) \cdot H\left(\hat{q_u}, p\left(y \mid u\right)\right)\right]. \end{aligned}$$

Here H is the loss function, τ is the threshold for pseudo label selection, q is the confidence distribution, $\hat{q}$ is the pseudo label, and $R(\theta)$ is the total loss on labeled dataset S and unlabeled dataset U. Now assume the weight of an unlabeled example u' is increased by ϵ. Denote $\mathcal{L}_U(u', \theta) = \lambda \cdot \mathbb{1}\left(\max(q_{u'}) \geq \tau\right) \cdot H\left(\hat{q_{u'}}, p\left(y \mid u'\right)\right)$, the optimal model parameters corresponding to the new training set become:

$$\hat{\theta} = \arg\min_{\theta} \; R(\theta) + \epsilon \cdot \mathcal{L}_U(u', \theta). \tag{17}$$

In Equation 17, $\hat{\theta}$ minimizes the loss function on the training set, which means the gradient w.r.t $\hat{\theta}$ is 0:

$$\nabla_\theta R(\hat{\theta}) + \epsilon \nabla_\theta \mathcal{L}_U(u', \hat{\theta}) = 0. \tag{18}$$

Using a Taylor-series approximation at θ^*, Equation 18 can be rewritten as:

$$\begin{aligned} & \nabla_\theta R\left(\theta^*\right) + \epsilon \cdot \nabla_\theta \mathcal{L}_U\left(u', \theta^*\right) \\ & + \left(\nabla_\theta^2 R\left(\theta^*\right) + \epsilon \cdot \nabla_\theta^2 \mathcal{L}_U\left(u', \theta^*\right)\right) \cdot \left(\hat{\theta} - \theta^*\right) = 0, \end{aligned} \tag{19}$$

which gives (please refer to Appendix B for details):

$$\begin{aligned} \hat{\theta} - \theta^* & \approx -\left(\nabla_\theta^2 R\left(\theta^*\right)\right)^{-1} \cdot \epsilon \nabla_\theta \mathcal{L}_U\left(u, \theta\right) \\ & \doteq -\epsilon \cdot H_\theta^{-1} \nabla_\theta \mathcal{L}_U\left(u, \theta\right). \end{aligned} \tag{20}$$

With the help of the chain rule $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\epsilon} = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\theta} \cdot \frac{\mathrm{d}\theta}{\mathrm{d}\epsilon}$, the importance of an unlabeled example can be estimated:

$$\begin{aligned} \mathrm{score}_\theta(u) = \frac{\mathrm{d}\mathcal{L}\left(V, \theta\right)}{\mathrm{d}\epsilon} & = \nabla_\theta \mathcal{L}\left(V, \theta\right)^\top \frac{\mathrm{d}\theta}{\mathrm{d}\epsilon} \\ & = -\nabla_\theta \mathcal{L}\left(V, \theta\right)^\top H_\theta^{-1} \nabla_\theta \mathcal{L}_U\left(u, \theta\right). \end{aligned} \tag{21}$$

Equation 21 is used to compute $\mathrm{score}_\theta(u)$ for each unlabeled example. The unlabeled examples with the highest score are preserved to build the core set, and others will be simply dropped. In our implementation, the INFUSE score is calculated batch-wise to reduce the computation overhead. Besides, we use the identity matrix to approximate the inverse Hessian H_θ^{-1} [Luketina *et al.*, 2016] for efficiency. The last problem is how to compute $\nabla_\theta \mathcal{L}\left(V, \theta\right)$ when the ground-truth label of examples in V is unavailable in training. To address this, we propose a feature-level mixup to build a support set $\overline{S}$. Then, the gradient on the validation set is approximated by $\mathcal{L}\left(\overline{S}, \theta\right)$. Please refer to Appendix C for details.

5 Experiments

5.1 Experiment Settings

We evaluate the effectiveness of our method on standard semi-supervised learning (SSL) datasets: CIFAR-10/100 [Krizhevsky *et al.*, 2009], SVHN [Netzer *et al.*, 2011], STL-10 [Coates *et al.*, 2011]. We follow the commonly used SSL setting [Sohn *et al.*, 2020] for model training. The keep ratio k controls the size of the core set. For example, with $k = 10\%$, the core set size is $10\% \times |U|$, and the total training steps become 10% of the original iterations.

The model is trained under the most commonly used SSL setting [Sohn *et al.*, 2020]. The total number of iterations is 2^{20} (segmented into 1024 epochs) and batch-size of labeled/unlabeled data is 64/448. We use SGD to optimize the parameters. The learning rate is initially set as $\eta_0 = 0.03$ with a cosine learning rate decay schedule as $\eta = \eta_0 \cos\left(\frac{7\pi k}{16K}\right)$, where k is the current iteration and K is the total iterations.

As for VCC, the size of random noise z is set as 16 for best performance. To reduce the computation overhead, the encoder q_ϕ and decoder p_θ are MLPs with 2 hidden layers (with dimensions 256 and 64). λ_{VCC} is set as 2.0.

In INFUSE, the core set is updated for every 40 epochs, and the total number of iterations is adjusted with the keep ratio k. Take $k = 10\%$ for example, the amount of examples in core set is $10\% \times |U|$ and the total steps is $10\% \times 2^{20}$.

5.2 Main Results

In this section, we present the effectiveness of VCC and INFUSE individually and then combine them to achieve more efficient and accurate pseudo label selection in SSL.

As mentioned earlier, VCC is a general confidence calibration plugin, allowing flexible combinations with existing SSL methods. In our experiments, we choose popular methods like FixMatch [Sohn *et al.*, 2020], FlexMatch [Zhang *et al.*, 2021], and SimMatch [Zheng *et al.*, 2022] as the basic modules to build VCC-FixMatch, VCC-FlexMatch, and VCC-SimMatch. The reported values are the mean and standard deviation of three independent trials for each setting, as shown

Method	CIFAR-10			CIFAR-100			SVHN		
	40	250	2500	400	2500	10000	40	250	1000
PL	76.29$_{\pm 1.08}$	48.28$_{\pm 2.01}$	14.90$_{\pm 0.20}$	87.15$_{\pm 0.47}$	59.09$_{\pm 0.61}$	38.86$_{\pm 0.09}$	75.95$_{\pm 3.39}$	16.60$_{\pm 1.13}$	9.33$_{\pm 0.58}$
UDA	8.01$_{\pm 1.34}$	5.12$_{\pm 0.15}$	4.32$_{\pm 0.07}$	53.44$_{\pm 2.06}$	34.37$_{\pm 0.28}$	27.52$_{\pm 0.10}$	2.03$_{\pm 0.02}$	2.03$_{\pm 0.03}$	1.96$_{\pm 0.01}$
VAT	76.42$_{\pm 2.57}$	42.58$_{\pm 6.67}$	10.97$_{\pm 0.19}$	83.11$_{\pm 0.27}$	53.17$_{\pm 0.57}$	36.58$_{\pm 0.21}$	77.00$_{\pm 6.59}$	4.59$_{\pm 0.13}$	4.09$_{\pm 0.21}$
MeanTeacher	76.93$_{\pm 2.29}$	56.06$_{\pm 2.03}$	15.47$_{\pm 0.43}$	90.34$_{\pm 0.65}$	61.13$_{\pm 0.57}$	39.05$_{\pm 0.12}$	81.94$_{\pm 1.33}$	25.10$_{\pm 3.17}$	12.29$_{\pm 0.45}$
MixMatch	70.67$_{\pm 1.25}$	37.28$_{\pm 0.61}$	7.38$_{\pm 0.06}$	79.95$_{\pm 0.29}$	49.58$_{\pm 0.62}$	32.10$_{\pm 0.13}$	79.63$_{\pm 5.78}$	3.71$_{\pm 0.20}$	3.12$_{\pm 0.09}$
ReMixMatch	14.50$_{\pm 2.58}$	9.21$_{\pm 0.55}$	4.89$_{\pm 0.05}$	57.10$_{\pm 0.01}$	34.77$_{\pm 0.32}$	26.18$_{\pm 0.23}$	31.27$_{\pm 18.79}$	6.38$_{\pm 1.09}$	5.34$_{\pm 0.45}$
Dash(RandAug)	15.01$_{\pm 3.70}$	5.13$_{\pm 0.26}$	4.35$_{\pm 0.09}$	53.98$_{\pm 2.31}$	34.47$_{\pm 0.12}$	27.72$_{\pm 0.03}$	2.08$_{\pm 0.09}$	1.97$_{\pm 0.01}$	2.03$_{\pm 0.03}$
SoftMatch	5.06$_{\pm 0.02}$	4.84$_{\pm 0.10}$	4.27$_{\pm 0.12}$	49.64$_{\pm 1.46}$	33.05$_{\pm 0.05}$	27.26$_{\pm 0.03}$	2.31$_{\pm 0.01}$	2.15$_{\pm 0.05}$	2.08$_{\pm 0.04}$
CoMatch	5.44$_{\pm 0.05}$	5.33$_{\pm 0.12}$	4.29$_{\pm 0.04}$	60.98$_{\pm 0.77}$	37.24$_{\pm 0.24}$	28.15$_{\pm 0.16}$	9.51$_{\pm 5.59}$	2.21$_{\pm 0.20}$	1.96$_{\pm 0.07}$
FixMatch	7.52$_{\pm 0.42}$	4.90$_{\pm 0.03}$	4.28$_{\pm 0.10}$	46.47$_{\pm 0.05}$	28.09$_{\pm 0.06}$	22.21$_{\pm 0.02}$	**2.96**$_{\pm 1.23}$	1.99$_{\pm 0.05}$	1.96$_{\pm 0.06}$
VCC-FixMatch	**6.84**$_{\pm 0.52}$	**4.68**$_{\pm 0.04}$	**4.27**$_{\pm 0.21}$	**43.31**$_{\pm 0.02}$	**27.76**$_{\pm 0.06}$	**22.05**$_{\pm 0.03}$	3.12$_{\pm 0.61}$	**1.97**$_{\pm 0.02}$	**1.95**$_{\pm 0.08}$
FlexMatch	4.98$_{\pm 0.01}$	5.00$_{\pm 0.05}$	4.24$_{\pm 0.07}$	40.43$_{\pm 0.63}$	26.38$_{\pm 0.17}$	21.83$_{\pm 0.08}$	3.36$_{\pm 0.37}$	5.02$_{\pm 1.20}$	5.43$_{\pm 0.46}$
VCC-FlexMatch	**4.90**$_{\pm 0.10}$	**4.65**$_{\pm 0.07}$	**4.14**$_{\pm 0.15}$	**37.98**$_{\pm 0.65}$	**25.75**$_{\pm 0.11}$	**21.48**$_{\pm 0.07}$	**2.62**$_{\pm 0.08}$	**4.97**$_{\pm 0.08}$	**3.71**$_{\pm 1.13}$
SimMatch	5.60$_{\pm 1.37}$	4.84$_{\pm 0.39}$	3.96$_{\pm 0.01}$	37.81$_{\pm 2.21}$	25.07$_{\pm 0.32}$	**20.58**$_{\pm 0.11}$	3.70$_{\pm 0.72}$	2.27$_{\pm 0.12}$	**2.07**$_{\pm 0.08}$
VCC-SimMatch	**5.27**$_{\pm 0.34}$	**4.76**$_{\pm 0.14}$	**3.87**$_{\pm 0.24}$	**37.22**$_{\pm 0.04}$	**24.98**$_{\pm 0.13}$	20.61$_{\pm 0.01}$	**3.04**$_{\pm 0.02}$	**2.20**$_{\pm 0.01}$	4.39$_{\pm 0.02}$
Fully-Supervised		4.58$_{\pm 0.05}$			19.63$_{\pm 0.08}$			2.07$_{\pm 0.02}$	

Table 1: Comparison of error rate (%) for different methods under various settings.

Labels	FixMatch	w/ VCC	FlexMatch	w/ VCC	SimMatch	w/ VCC
40	35.97	**30.63**	29.15	**28.14**	27.84	**26.97**
1000	6.25	**5.31**	5.77	**5.52**	5.91	**5.51**

Table 2: Error rate results (%) on STL-10 dataset.

in Table 1. All three baseline methods (FixMatch, FlexMatch, SimMatch) exhibit accuracy improvements when combined with VCC for confidence calibration. Particularly, the improvements with VCC are more pronounced when the amount of labeled examples is small. For instance, on CIFAR-100 with only 400 labeled examples, VCC-FlexMatch reduces the error rate from 46.47% to 43.31% (-3.16%). A similar boost is observed on the STL-10 dataset, as shown in Table 2, where VCC reduces the error rate of FixMatch by 5.34% (from 35.97% to 30.63%) with only 40 labels.

To further understand the source of the accuracy improvement with VCC, we compute the calibration error of different methods. As shown in Table 3, both VCC-FixMatch and VCC-FlexMatch achieve lower calibration errors compared to the baseline methods across various settings. VCC-SimMatch also achieves lower Expected Calibration Error (ECE) and Average Calibration Error (ACE) metrics when only 400 labeled examples are available. However, the Maximum Calibration Error (MCE) metric deteriorates, attributed to MCE considering the worst-calibrated bucket and introducing some fluctuations. Under the setting of using 10,000 labeled examples, the results of VCC-SimMatch and SimMatch are very close. This is partly because a larger number of labeled examples can naturally improve the model's performance and reduce the calibration error. Additionally, SimMatch uses instance similarity for rescaling the confidence score, which may reduce the benefits brought by VCC.

The results of INFUSE and other core set selection methods (e.g., RETRIEVE [Killamsetty *et al.*, 2021b]) are shown in Table 4. On CIFAR-10 dataset, INFUSE achieves a lower error rate (6.29%) using only 10% of the examples, indicating the redundancy of original unlabeled data and underscoring the significance of core set selection in SSL. With increasing keep ratio, the gap between INFUSE and the non-pruned setting becomes smaller. For example, on the CIFAR-100 dataset with 2500 labeled data and a keep ratio of 40%, INFUSE achieves an error rate of 26.47%, while the baseline is 26.49%. Compared with other core set selection methods, INFUSE also achieves lower error rates in most settings.

The results demonstrate the effectiveness of VCC and INFUSE individually. By combining them, we propose the VCC-INFUSE method, with results shown in Table 5. VCC-INFUSE achieves a better trade-off between model performance and computation costs. Compared to FlexMatch, VCC-INFUSE-FlexMatch not only reduces the error rate from 26.49% to 25.41% but also decreases the training time from 223.96 GPU Hours to 115.47 GPU Hours (-48.44%).

5.3 Ablation Study

We utilize view consistency, temporal consistency, and ensemble consistency for estimating $\tilde{r}$. These three consistency scores are designed to reflect the stability of predictions from different perspectives. To analyze their contributions, we conduct an ablation study, and the result is shown in Table 7. As observed, each consistency score contributes to the estimation of a more accurate $\tilde{r}$, resulting in a lower error rate.

5.4 Effectiveness of VCC

In VCC, we initially approximate calibrated confidence to obtain $\tilde{r}_u$. Subsequently, we use a Variational Autoencoder (VAE) to reconstruct it, yielding r_u, which is employed in pseudo label selection. The objective of reconstruction is to mitigate the randomness associated with statistical approximation. To demonstrate its necessity, we conduct an ablation study. As shown in Table 8, VCC with reconstruction further reduces the error rate by 0.50%.

5.5 VCC vs. Other Calibration Methods

While most calibration methods designed for fully-supervised settings may not be directly suitable for SSL, pseudo labels can be used to approximate ground truth. We

Method	400 labels				2500 labels				10000 labels			
	ER(%)	ECE	MCE	ACE	ER(%)	ECE	MCE	ACE	ER(%)	ECE	MCE	ACE
FixMatch	46.42	0.382	0.573	0.376	28.03	0.208	0.530	0.199	22.20	0.127	0.322	0.128
VCC-FixMatch	**43.29**	**0.359**	**0.560**	**0.345**	**27.81**	**0.195**	**0.418**	**0.182**	**22.01**	**0.125**	**0.317**	**0.127**
FlexMatch	39.94	0.291	0.512	0.286	26.49	0.169	0.369	0.173	21.90	0.120	0.311	0.126
VCC-FlexMatch	**37.52**	**0.257**	**0.446**	**0.258**	**25.26**	**0.147**	**0.324**	**0.163**	**21.55**	**0.104**	**0.269**	**0.125**
SimMatch	37.81	0.325	**0.510**	0.328	25.07	0.157	0.358	0.179	**20.58**	**0.113**	0.295	**0.116**
VCC-SimMatch	**37.20**	**0.317**	0.514	**0.314**	**25.01**	**0.155**	**0.347**	**0.173**	20.61	0.115	**0.291**	0.121

Table 3: Error rate, ECE [Guo *et al.*, 2017], MCE [Guo *et al.*, 2017], and ACE [Nixon *et al.*, 2019] results on CIFAR-100 with different number of labeled exmaples.

Method	**CIFAR-10**					**CIFAR-100**					**STL-10**		**SVHN**	
	250 label			**4000 label**		**2500 label**			**10000 label**		**250 label**		**250 label**	
	10%	20%	40%	40%	60%	10%	20%	40%	40%	60%	10%	20%	10%	20%
Random	9.12	6.87	6.51	5.26	5.01	31.55	31.11	28.86	23.19	22.51	16.62	14.37	3.85	4.65
Earlystop	7.47	6.03	6.85	4.86	4.52	29.21	28.85	27.30	23.03	22.61	16.31	13.20	2.93	3.08
EL2N	8.55	7.47	6.70	4.94	4.54	31.55	31.27	28.42	23.12	22.21	16.27	12.92	3.66	3.61
GradMatch	6.71	5.87	5.60	4.72	4.45	28.95	28.48	26.71	22.72	22.21	16.05	12.90	2.90	2.63
RETRIEVE	6.60	6.02	5.48	4.68	4.41	**28.75**	28.34	26.68	22.56	22.18	16.05	12.90	2.90	2.63
INFUSE (Ours)	**6.29**	**5.69**	**5.33**	**4.51**	**4.34**	28.83	**28.05**	**26.47**	**22.28**	**21.97**	**15.84**	**12.71**	**2.61**	**2.46**
Full Unlabeled Data	4.98			4.19		26.49			21.90		8.23		3.80	

Table 4: Comparison of error rate (%) for core set selection on different datasets and example keep raito (from 10% to 60%).

Method	Error Rate (%)	Training time
Dash(RandAug)	27.15	-
MPL	27.71	-
FixMatch	28.03	221.91
FlexMatch	26.49	223.96
VCC-FlexMatch (Ours)	**25.26**	253.53
VCC-INFUSE-FlexMatch (Ours, keep raito=40%)	25.41	**115.47**

Table 5: The error rate and GPU Hours on A100 of different methods on CIFAR-100 dataset with 2500 labeled data.

Method	ER(%)	ECE	MCE	ACE
FlexMatch	26.49	0.169	0.369	0.173
FlexMatch + Ensemble-TS [Zhang *et al.*, 2020]	26.36	0.165	0.382	0.174
FlexMatch + MMCE [Kumar *et al.*, 2018]	28.44	0.182	0.374	0.185
VCC-FlexMatch (Ours)	**25.26**	**0.147**	**0.324**	**0.163**

Table 6: The error rate of VCC and other calibration methods on CIFAR-100 dataset with 2500 labeled examples.

ensemble	**temporal**	**vieW**	**ER(%)**	**ECE**	**MCE**	**ACE**
✓	✓	✗	25.45	0.148	0.328	0.167
✓	✗	✓	25.97	0.166	0.352	0.168
✗	✓	✓	25.65	0.153	0.337	0.169
✓	✓	✓	**25.26**	**0.147**	**0.324**	**0.163**

Table 7: The Error Rate (ER) and calibration errors of VCC when different consistency score is disabled while approximating the $\tilde{r}_u$. Tested under CIFAR-100 dataset with 2500 labeled examples.

Reconstruct $\tilde{r}_u$ by VAE	**Error Rate (%)**	**ECE**	**MCE**	**ACE**
✗	25.76	0.160	0.411	0.168
✓	**25.26**	**0.147**	**0.324**	**0.163**

Table 8: Error rates of VCC with or without reconstructing calibrated confidence, on CIFAR-100 with 2500 labeled examples.

choose Ensemble-TS [Zhang *et al.*, 2020] and MMCE [Kumar *et al.*, 2018] as baselines to compare with VCC. As depicted in Table 6, MMCE exhibits the highest error rate (28.44%). The reason is that MMCE directly employs pseudo labels to calculate calibration regularization, which may introduce noise due to incorrect pseudo labels. In contrast, Ensemble-TS, using pseudo labels to search for optimal parameter scaling, alleviates the issue to some extent (ER=26.36%). In comparison, VCC achieves the lowest error rate of 25.26% and the best calibration performance.

6 Conclusion

In this study, we addressed the challenges associated with leveraging large-scale unlabeled data in SSL and proposed two novel methods, VCC and INFUSE, to enhance the effectiveness and efficiency of data selection. As a versatile plugin, VCC significantly improves the accuracy of FixMatch, FlexMatch, and SimMatch across multiple datasets. Simultaneously, INFUSE achieves competitive or even lower error rates with partial unlabeled data. By combining the two methods, VCC-INFUSE achieves a lower error rate with less computational overhead. Future work will involve extending VCC-INFUSE to various SSL tasks, such as object detection and segmentation, to assess its generalization.

Acknowledgements

This work was supported by National Key R&D Program of China (No.2018AAA0100300) and the funding of China Tower.

Contribution Statement

Shijie Fang and Qianhan Feng contribute equally to this paper. Tong Lin is the corresponding author. The work was done during Shijie Fang's study at Peking University.

References

[Berthelot *et al.*, 2020] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring. In *International Conference on Learning Representations, Addis Ababa, Ethiopia*, 2020.

[Chen *et al.*, 2023] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality tradeoff in semi-supervised learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[Coates *et al.*, 2011] Adam Coates, Andrew Y. Ng, and Honglak Lee. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, USA*, 2011.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition, Miami, Florida, USA*, 2009.

[Feng *et al.*, 2024] Qianhan Feng, Lujing Xie, Shijie Fang, and Tong Lin. Bacon: Boosting imbalanced semi-supervised learning via balanced feature-level contrastive learning. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024*, 2024.

[Gal and Ghahramani, 2016] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33nd International Conference on Machine Learning, New York City, NY, USA*, 2016.

[Ganev and Aitchison, 2020] Stoil Ganev and Laurence Aitchison. Semi-supervised Learning Objectives as Log-likelihoods in a Generative Model of Data Curation. *arXiv preprint arXiv:2008.05913*, 2020.

[Guo *et al.*, 2017] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning, Sydney, NSW, Australia*, 2017.

[Killamsetty *et al.*, 2021a] KrishnaTeja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh K. Iyer. GRAD-MATCH: Gradient Matching based Data Subset Selection for Efficient Deep Model Training. In *Proceedings of the International Conference on Machine Learning, Virtual Event*, 2021.

[Killamsetty *et al.*, 2021b] KrishnaTeja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh K. Iyer. RETRIEVE: Coreset Selection for Efficient and Robust Semi-Supervised Learning. In *Advances in Neural Information Processing Systems, virtual*, 2021.

[Kingma and Welling, 2014] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations, Banff, AB, Canada*, 2014.

[Koh and Liang, 2017] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning Multiple Layers of Features from Tiny Images. In *Doctoral dissertation, University of Toronto*, 2009.

[Kumar *et al.*, 2018] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable Calibration Measures For Neural Networks From Kernel Mean Embeddings. In *International Conference on Machine Learning, Stockholm, Sweden*, 2018.

[Laine and Aila, 2016] Samuli Laine and Timo Aila. Temporal Ensembling for Semi-Supervised Learning. *arXiv preprint arXiv:1610.02242*, 2016.

[Lee *et al.*, 2021] Hyuck Lee, Seungjae Shin, and Heeyoung Kim. ABC: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2021.

[Li *et al.*, 2021] Junnan Li, Caiming Xiong, and Steven C. H. Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, October 10-17*, 2021.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision, Zurich, Switzerland*, 2014.

[Luketina *et al.*, 2016] Jelena Luketina, Tapani Raiko, Mathias Berglund, and Klaus Greff. Scalable Gradient-Based Tuning of Continuous Regularization Hyperparameters. In *International Conference on Machine Learning, New York City, NY, USA*, 2016.

[Miyato *et al.*, 2019] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2019.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[Nixon *et al.*, 2019] Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring Calibration in Deep Learning. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA*, 2019.

[Paul *et al.*, 2021] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep Learning on a Data Diet: Finding Important Examples Early in Training. In *Advances in Neural Information Processing Systems, virtual*, 2021.

[Sohn *et al.*, 2020] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *Advances in Neural Information Processing Systems, Virtual*, 2020.

[Xu *et al.*, 2015] Chang Xu, Dacheng Tao, and Chao Xu. Multi-View Learning With Incomplete Views. *IEEE Trans. Image Process.*, 24(12):5812–5825, 2015.

[Zhang *et al.*, 2020] Jize Zhang, Bhavya Kailkhura, and Thomas Yong-Jin Han. Mix-N-Match : Ensemble and Compositional Methods for Uncertainty Calibration in Deep Learning. In *International Conference on Machine Learning, Virtual*, 2020.

[Zhang *et al.*, 2021] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling. In *Advances in Neural Information Processing, Virtual*, 2021.

[Zheng *et al.*, 2022] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. SimMatch: Semi-supervised Learning with Similarity Matching. In *Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA*, 2022.

[Zhou *et al.*, 2020] Tianyi Zhou, Shengjie Wang, and Jeff A. Bilmes. Time-Consistent Self-Supervision for Semi-Supervised Learning. In *International Conference on Machine Learning, Virtual*, 2020.