Theory-Inspired Contrastive Learning with Self-Labeling Refinement

Pan Zhou, Caiming Xiong, Xiaotong Yuan, and Steven HOI

Salesforce

panzhou3@gmail.com

Dec 06, 2021

Outline

Motivation: why one-hot label in MoCo is not accurate?

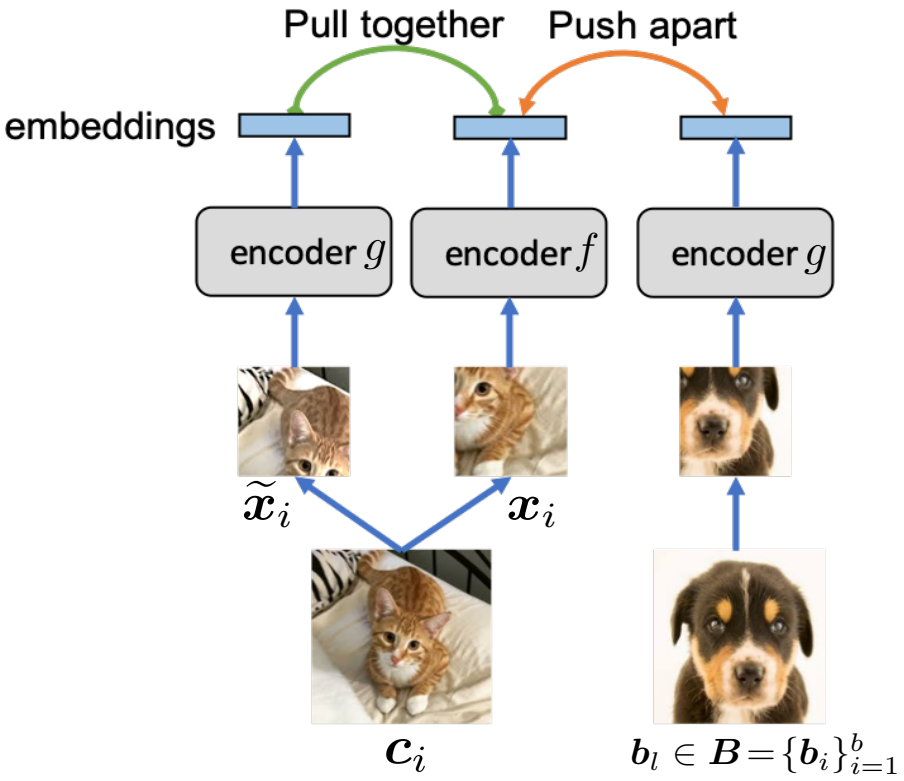
Solution for accurate label: self-labeling refinery and momentum mixup

Experiments: higher classification accuracy

Conclusion

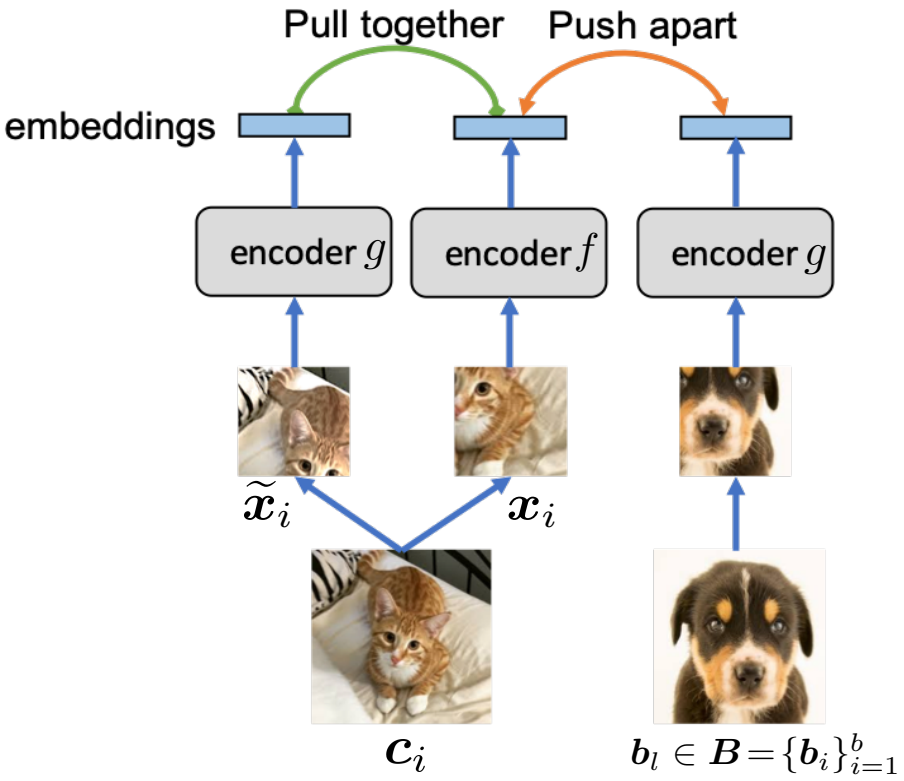
Background: MoCo

MoCo pulls together crops of same image, pushs away crops of different images.



Background: MoCo

MoCo pulls together crops of same image, pushs away crops of different images.



Given a minibatch samples $\{\boldsymbol{c}_i\}_{i=1}^s$, it augments each $\boldsymbol{c}_i$ into two views $(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)$ and optimizes:

$$\mathcal{L}_{\mathrm{n}}(\boldsymbol{w}) = -\frac{1}{s}\sum_{i=1}^{s} \log\left(\frac{\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)}{\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i) + \sum_{l=1}^{b}\sigma(\boldsymbol{x}_i, \boldsymbol{b}_l)}\right),$$

query positive key negative key

- $\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i) = \exp\left(-\frac{\langle f(\boldsymbol{x}_i), g(\widetilde{\boldsymbol{x}}_i)\rangle}{\tau \|f(\boldsymbol{x}_i)\|_2 \cdot \|g(\widetilde{\boldsymbol{x}}_i)\|_2}\right)$: cosine similarity function
- $f_{\boldsymbol{w}}$: online network
- $g_{\boldsymbol{\xi}}$: target network
- $\boldsymbol{B} = \{\boldsymbol{b}_i\}_{i=1}^b$: negative key buffer (previous minibatch crops)

Motivation from Observations

One-hot label assignment: query $\boldsymbol{x}_i$ has only one positive $\widetilde{\boldsymbol{x}}_i$ among $\widetilde{\boldsymbol{x}}_i \cup \boldsymbol{B}$

$$\mathcal{L}_{\mathrm{n}}(\boldsymbol{w}) = -\frac{1}{s}\sum_{i=1}^{s} \log\Big(\frac{\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)}{\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i) + \sum_{l=1}^{b}\sigma(\boldsymbol{x}_i, \boldsymbol{b}_l)}\Big),$$

query positive key negative key

Motivation from Observations

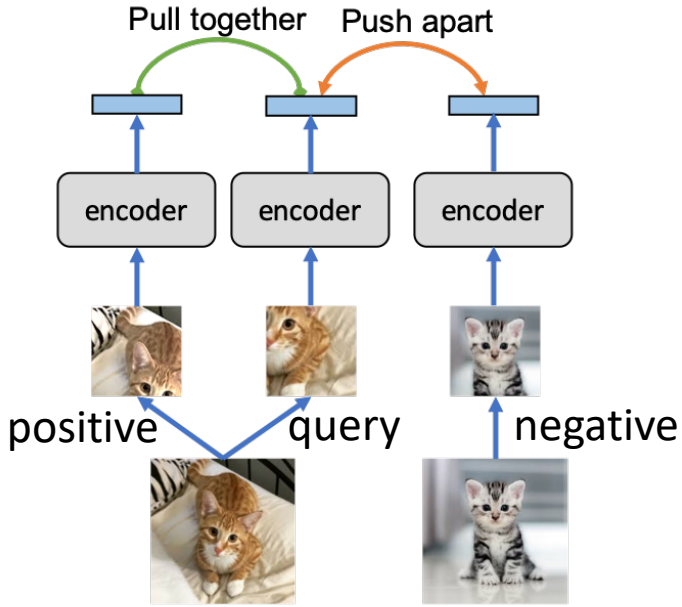
One-hot label assignment: query $\boldsymbol{x}_i$ has only one positive $\widetilde{\boldsymbol{x}}_i$ among $\widetilde{\boldsymbol{x}}_i \cup \boldsymbol{B}$

$$\mathcal{L}_{\mathrm{n}}(\boldsymbol{w}) = -\frac{1}{s}\sum_{i=1}^{s} \log\Big(\frac{\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)}{\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i) + \sum_{l=1}^{b}\sigma(\boldsymbol{x}_i, \boldsymbol{b}_l)}\Big),$$

query positive key negative key

Issues of hot label assignment: **imprecise & uninformative**

- some negatives & query belong to same semantic class



$\boldsymbol{b}_l \in \boldsymbol{B} = \{\boldsymbol{b}_i\}_{i=1}^{b}$

Motivation from Observations

One-hot label assignment: query $\boldsymbol{x}_i$ has only one positive $\widetilde{\boldsymbol{x}}_i$ among $\widetilde{\boldsymbol{x}}_i \cup \boldsymbol{B}$

$$\mathcal{L}_{\mathrm{n}}(\boldsymbol{w}) = -\frac{1}{s}\sum_{i=1}^{s} \log\Big(\frac{\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)}{\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i) + \sum_{l=1}^{b}\sigma(\boldsymbol{x}_i, \boldsymbol{b}_l)}\Big),$$

query positive key negative key

Issues of hot label assignment: imprecise and and uninformative

- some negatives & query belong to same semantic class

- random augmentations provides crops with different semantic information, e.g. image having several objects

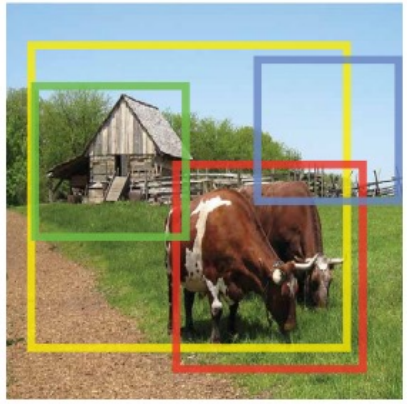


image in ImageNet

Random crop

cow house sky ...

Motivation from Observations

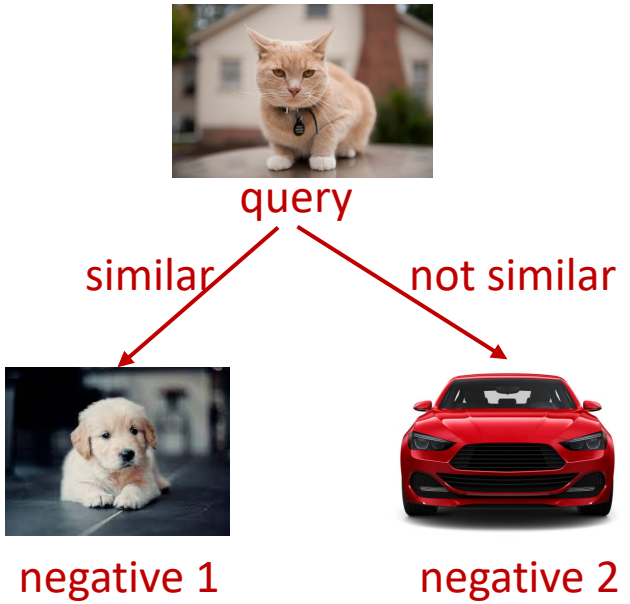
One-hot label assignment: query $\boldsymbol{x}_i$ has only one positive $\widetilde{\boldsymbol{x}}_i$ among $\widetilde{\boldsymbol{x}}_i \cup \boldsymbol{B}$

$$\mathcal{L}_{\mathrm{n}}(\boldsymbol{w}) = -\frac{1}{s}\sum_{i=1}^{s} \log\Big(\frac{\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)}{\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i) + \sum_{l=1}^{b}\sigma(\boldsymbol{x}_i, \boldsymbol{b}_l)}\Big),$$

query positive key negative key

Issues of hot label assignment: imprecise and and uninformative

- some negatives & query belong to same semantic class
- random augmentations provides crops with different semantic information, e.g. image having several objects
- different negatives have different similarity to query



Motivation from Observations

One-hot label assignment: query $\boldsymbol{x}_i$ has only one positive $\widetilde{\boldsymbol{x}}_i$ among $\widetilde{\boldsymbol{x}}_i \cup \boldsymbol{B}$

$$\mathcal{L}_{\mathrm{n}}(\boldsymbol{w}) = -\frac{1}{s}\sum_{i=1}^{s} \log\Big(\frac{\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)}{\sigma(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i) + \sum_{l=1}^{b}\sigma(\boldsymbol{x}_i, \boldsymbol{b}_l)}\Big),$$

query positive key negative key

Issues of hot label assignment: imprecise and and uninformative

- some negatives & query belong to same semantic class
- random augmentations provides crops with different semantic information, e.g. image having several objects
- different negatives have different similarity to query

Result: one-hot label cannot guarantee semantically similar samples to close

Motivation from Theoretical Analysis

Support that the pair $(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)$ in the training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)\}_{i=1}^n$ sampled from an unknown distribution $\mathcal{S}$ denotes the positive pair in MoCo.

Assume the query $\boldsymbol{x}_i$ has ground truth soft label $\boldsymbol{y}_i^* \in b+1$ over the key set $\boldsymbol{B}_i = \{\widetilde{\boldsymbol{x}}_i \cup \boldsymbol{B}\}$ where $\boldsymbol{y}_{it}^*$ measures the semantic similarity between $\boldsymbol{x}_i$ and the t-th key $\boldsymbol{b}_t'$ in buffer $\boldsymbol{B}_i$

Theorem 1 (upper bound of generalization error, informal).

Under proper assumptions, for MoCo, with probability $1-\nu$, the generalization error on instance discrimination task can be upper bounded as :

$$\text{generalization error} \leq \mathcal{O}\left(\mathbb{E}_{\mathcal{D}\sim\mathcal{S}}\left[\|\boldsymbol{y}-\boldsymbol{y}^*\|_2\right]\right) + \mathcal{O}\left(\sqrt{\frac{V_{\mathcal{D}}\ln(|\mathcal{F}|/\nu)}{n}} + \frac{\ln(|\mathcal{F}|/\nu)}{n}\right),$$

training & test error gap

where $V_{\mathcal{D}}$ is the variance of $f_{\boldsymbol{w}}$ on data $\mathcal{D}$, $\mathcal{F}$ is the covering number of encoder $f_{\boldsymbol{w}}$

Motivation from Theoretical Analysis

Support that the pair $(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)$ in the training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)\}_{i=1}^n$ sampled from an unknown distribution $\mathcal{S}$ denotes the positive pair in MoCo.

Assume the query $\boldsymbol{x}_i$ has ground truth soft label $\boldsymbol{y}_i^* \in b+1$ over the key set $\boldsymbol{B}_i = \{\widetilde{\boldsymbol{x}}_i \cup \boldsymbol{B}\}$ where $\boldsymbol{y}_{it}^*$ measures the semantic similarity between $\boldsymbol{x}_i$ and the t-th key $\boldsymbol{b}_t'$ in buffer $\boldsymbol{B}_i$

Theorem 1 (upper bound of generalization error, informal).

Under proper assumptions, for MoCo, with probability $1-\nu$, the generalization error on instance discrimination task can be upper bounded as :

one-hot label　　true soft label

$$\text{generalization error} \leq \mathcal{O}\left(\mathbb{E}_{\mathcal{D}\sim\mathcal{S}}\left[\|\boldsymbol{y}-\boldsymbol{y}^*\|_2\right]\right) + \mathcal{O}\left(\sqrt{\frac{V_{\mathcal{D}}\ln(|\mathcal{F}|/\nu)}{n}} + \frac{\ln(|\mathcal{F}|/\nu)}{n}\right),$$

training & test error gap

where $V_{\mathcal{D}}$ is the variance of $f_{\boldsymbol{w}}$ on data $\mathcal{D}$, $\mathcal{F}$ is the covering number of encoder $f_{\boldsymbol{w}}$

Motivation from Theoretical Analysis

Support that the pair $(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)$ in the training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)\}_{i=1}^n$ sampled from an unknown distribution $\mathcal{S}$ denotes the positive pair in MoCo.

Assume the query $\boldsymbol{x}_i$ has ground truth soft label $\boldsymbol{y}_i^* \in b+1$ over the key set $\boldsymbol{B}_i = \{\widetilde{\boldsymbol{x}}_i \cup \boldsymbol{B}\}$ where $\boldsymbol{y}_{it}^*$ measures the semantic similarity between $\boldsymbol{x}_i$ and the t-th key $\boldsymbol{b}_t'$ in buffer $\boldsymbol{B}_i$

Theorem 1 (upper bound of generalization error, informal).

Under proper assumptions, for MoCo, with probability $1-\nu$, the generalization error on instance discrimination task can be upper bounded as :

one-hot label　　true soft label

$$\text{generalization error} \leq \mathcal{O}\left(\mathbb{E}_{\mathcal{D}\sim\mathcal{S}}\left[\|\boldsymbol{y} - \boldsymbol{y}^*\|_2\right]\right) + \mathcal{O}\left(\sqrt{\frac{V_{\mathcal{D}}\ln(|\mathcal{F}|/\nu)}{n}} + \frac{\ln(|\mathcal{F}|/\nu)}{n}\right),$$

training & test error gap　　the more accurate of the label y, the better the generalization

where $V_{\mathcal{D}}$ is the variance of $f_{\boldsymbol{w}}$ on data $\mathcal{D}$, $\mathcal{F}$ is the covering number of encoder $f_{\boldsymbol{w}}$

Motivation from Theoretical Analysis

Support that the pair $(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)$ in the training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)\}_{i=1}^n$ sampled from an unknown distribution $\mathcal{S}$ denotes the positive pair in MoCo.

Assume the query $\boldsymbol{x}_i$ has ground truth soft label $\boldsymbol{y}_i^* \in b+1$ over the key set $\boldsymbol{B}_i = \{\widetilde{\boldsymbol{x}}_i \cup \boldsymbol{B}\}$ where $\boldsymbol{y}_{it}^*$ measures the semantic similarity between $\boldsymbol{x}_i$ and the t-th key $\boldsymbol{b}_t'$ in buffer $\boldsymbol{B}_i$

Theorem 2 (lower bound of generalization error, informal).

Under proper assumptions, for MoCo, there exists a contrastive learning problem such that the generalization error on instance discrimination task is lower bounded as :

one-hot label true soft label

$$\text{generalization error} \geq \mathcal{O}\left(\mathbb{E}_{\mathcal{D}\sim\mathcal{S}}\left[\|\boldsymbol{y} - \boldsymbol{y}^*\|_2\right]\right).$$

Lower and upper bounds show that $\text{generalization error} \sim \mathbb{E}_{\mathcal{D}\sim\mathcal{S}}\left[\|\boldsymbol{y} - \boldsymbol{y}^*\|_2\right]$.

the more accurate of the label y, the better the generalization

Outline

Motivation: why one-hot label in MoCo is not accurate?

Solution for accurate label: self-labeling refinery and momentum mixup

Experiments: higher classification accuracy

Conclusion

Solution: Self-Labeling Refinement

Self-Labeling Refinement :

- **self-labeling refinery:** soft label replaces one-hot label to directly improve label accuracy
- **momentum mixup:** increase similarity of positive pair to indirectly improve label accuracy

Solution: Self-Labeling Refinement

Self-Labeling Refinement :

- **self-labeling refinery:** soft label replaces one-hot label to directly improve label accuracy
- **momentum mixup:** increase similarity of positive pair to indirectly improve label accuracy

Reformulation of MoCo: for query $\boldsymbol{x}_i$ in minibatch $\{(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)\}_{i=1}^{s}$, we maximize its similarity to its positive $\widetilde{\boldsymbol{x}}_i$ in the key set $\bar{\boldsymbol{B}} = \{\widetilde{\boldsymbol{x}}_i\}_{i=1}^{s} \cup \{\boldsymbol{b}_i\}_{i=1}^{b}$ and push it away from samples in $\bar{\boldsymbol{B}}$:

$$\mathcal{L}_{\mathrm{c}}\left(\boldsymbol{w}, \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}\right) = -\frac{1}{s}\sum_{i=1}^{s}\sum_{k=1}^{s+b} \boldsymbol{y}_{ik} \log\left(\frac{\sigma(\boldsymbol{x}_i, \bar{\boldsymbol{b}}_k)}{\sum_{l=1}^{s+b}\sigma(\boldsymbol{x}_i, \bar{\boldsymbol{b}}_l)}\right),$$

where $\bar{\boldsymbol{b}}_k$ is the k-th sample in $\bar{\boldsymbol{B}}$, $\boldsymbol{y}_i$ is the one-hot label of query $\boldsymbol{x}_i$ whose i-th entry $\boldsymbol{y}_{ii}$ is 1.

Solution: Self-Labeling Refinement

Self-Labeling Refinement :

- **self-labeling refinery:** soft label replaces one-hot label to directly improve label accuracy
- **momentum mixup:** increase similarity of positive pair to indirectly improve label accuracy

Reformulation of MoCo: for query $\boldsymbol{x}_i$ in minibatch $\{(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)\}_{i=1}^{s}$, we maximize its similarity to its positive $\widetilde{\boldsymbol{x}}_i$ in the key set $\bar{\boldsymbol{B}} = \{\widetilde{\boldsymbol{x}}_i\}_{i=1}^{s} \cup \{\boldsymbol{b}_i\}_{i=1}^{b}$ and push it away from samples in $\bar{\boldsymbol{B}}$:

$$\mathcal{L}_{\mathrm{c}}\left(\boldsymbol{w}, \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}\right) = -\frac{1}{s}\sum_{i=1}^{s}\sum_{k=1}^{s+b} \boldsymbol{y}_{ik} \log\left(\frac{\sigma(\boldsymbol{x}_i, \bar{\boldsymbol{b}}_k)}{\sum_{l=1}^{s+b}\sigma(\boldsymbol{x}_i, \bar{\boldsymbol{b}}_l)}\right),$$

where $\bar{\boldsymbol{b}}_k$ is the k-th sample in $\bar{\boldsymbol{B}}$, $\boldsymbol{y}_i$ is the one-hot label of query $\boldsymbol{x}_i$ whose i-th entry $\boldsymbol{y}_{ii}$ is 1.

Benefit of Reformulation: labels of different samples are defined on a shared dictionary, and thus can be linearly combined.

Solution: Self-Labeling Refinement

Self-Labeling Refinement : (1) self-labeling refinery; (2) momentum mixup

Self-labeling refinery iteratively employs network and data to improve labels during training.

- Step1. for query $\boldsymbol{x}_i$, we use its positive $\widetilde{\boldsymbol{x}}_i$ to estimate semantic similarity between $\boldsymbol{x}_i$ and instances in $\bar{\boldsymbol{B}} = \{\widetilde{\boldsymbol{x}}_i\}_{i=1}^{s} \cup \{\boldsymbol{b}_i\}_{i=1}^{b}$, since $\boldsymbol{x}_i$ and $\widetilde{\boldsymbol{x}}_i$ come from the same image:

$$\boldsymbol{p}_{ik}^{t} = \sigma^{1/\tau'}(\widetilde{\boldsymbol{x}}_i, \bar{\boldsymbol{b}}_k) \Big/ \sum_{l=1}^{s+b} \sigma^{1/\tau'}(\widetilde{\boldsymbol{x}}_i, \bar{\boldsymbol{b}}_l),$$

Solution: Self-Labeling Refinement

Self-Labeling Refinement : (1) self-labeling refinery; (2) momentum mixup

Self-labeling refinery iteratively employs network and data to improve labels during training.

- Step1. for query $\boldsymbol{x}_i$, we use its positive $\widetilde{\boldsymbol{x}}_i$ to estimate semantic similarity between $\boldsymbol{x}_i$ and instances in $\bar{\boldsymbol{B}} = \{\widetilde{\boldsymbol{x}}_i\}_{i=1}^{s} \cup \{\boldsymbol{b}_i\}_{i=1}^{b}$, since $\boldsymbol{x}_i$ and $\widetilde{\boldsymbol{x}}_i$ come from the same image:

$$\boldsymbol{p}_{ik}^{t} = \sigma^{1/\tau'}(\widetilde{\boldsymbol{x}}_i, \bar{\boldsymbol{b}}_k) \Big/ \sum_{l=1}^{s+b} \sigma^{1/\tau'}(\widetilde{\boldsymbol{x}}_i, \bar{\boldsymbol{b}}_l),$$

- Step2. as $\widetilde{\boldsymbol{x}}_i$ is highly similar to itself in $\bar{\boldsymbol{B}}$, $\boldsymbol{p}_{ii}^{t}$ will be much larger than others and conceals the similarity of other semantically similar instances in $\bar{\boldsymbol{B}}$. So we remove $\widetilde{\boldsymbol{x}}_i$ from $\bar{\boldsymbol{B}}$

$$\boldsymbol{q}_{ik}^{t} = \sigma^{1/\tau'}(\widetilde{\boldsymbol{x}}_i, \bar{\boldsymbol{b}}_k) \Big/ \sum_{l=1, l\neq i}^{s+b} \sigma^{1/\tau'}(\widetilde{\boldsymbol{x}}_i, \bar{\boldsymbol{b}}_l), \quad \boldsymbol{q}_{ii}^{t} = 0.$$

Solution: Self-Labeling Refinement

Self-Labeling Refinement : (1) self-labeling refinery; (2) momentum mixup

Self-labeling refinery iteratively employs network and data to improve labels during training.

- Step1. for query $\boldsymbol{x}_i$, we use its positive $\widetilde{\boldsymbol{x}}_i$ to estimate semantic similarity between $\boldsymbol{x}_i$ and instances in $\bar{\boldsymbol{B}} = \{\widetilde{\boldsymbol{x}}_i\}_{i=1}^{s} \cup \{\boldsymbol{b}_i\}_{i=1}^{b}$, since $\boldsymbol{x}_i$ and $\widetilde{\boldsymbol{x}}_i$ come from the same image:

$$\boldsymbol{p}_{ik}^{t} = \sigma^{1/\tau'}(\widetilde{\boldsymbol{x}}_i, \bar{\boldsymbol{b}}_k) \Big/ \sum_{l=1}^{s+b} \sigma^{1/\tau'}(\widetilde{\boldsymbol{x}}_i, \bar{\boldsymbol{b}}_l),$$

- Step2. as $\widetilde{\boldsymbol{x}}_i$ is highly similar to itself in $\bar{\boldsymbol{B}}$, $\boldsymbol{p}_{ii}^{t}$ will be much larger than others and conceals the similarity of other semantically similar instances in $\bar{\boldsymbol{B}}$. So we remove $\widetilde{\boldsymbol{x}}_i$ from $\bar{\boldsymbol{B}}$

$$\boldsymbol{q}_{ik}^{t} = \sigma^{1/\tau'}(\widetilde{\boldsymbol{x}}_i, \bar{\boldsymbol{b}}_k) \Big/ \sum_{l=1, l\neq i}^{s+b} \sigma^{1/\tau'}(\widetilde{\boldsymbol{x}}_i, \bar{\boldsymbol{b}}_l), \quad \boldsymbol{q}_{ii}^{t} = 0.$$

Linear combination:

$$\bar{\boldsymbol{y}}_i^{t} = (1 - \alpha_t - \beta_t)\boldsymbol{y}_i + \alpha_t \boldsymbol{p}_i^{t} + \beta_t \boldsymbol{q}_i^{t},$$

Performance Guarantee: Exact label recovery

Label-corrupted dataset

Let $\{(\boldsymbol{x}_i, \boldsymbol{y}_i^*)\}_{i=1}^n$ denote the pairs of crops and ground-truth semantic label

- crop $\boldsymbol{x}_i$ generated from vanilla sample $\boldsymbol{c}_t$ obeys $\|\boldsymbol{x}_i - \boldsymbol{c}_t\|_2 \leq \varepsilon$
- ground-truth semantic label $\boldsymbol{y}_i^* \in \{\gamma_t\}_{t=1}^{\bar{K}}$ of $\boldsymbol{x}_i$ is decided by its corresponding
- the classes are separated: $|\gamma_i - \gamma_k| \geq \delta, \quad \|\boldsymbol{c}_i - \boldsymbol{c}_k\|_2 \geq 2\varepsilon, \quad (\forall i \neq k),$
- for each sample $\boldsymbol{c}_i$, at most ρn_i augmentations are assigned to wrong labels, where n_i denotes the crop sample number of $\boldsymbol{c}_i$

Performance Guarantee: Exact label recovery

Label-corrupted dataset

Let $\{(\boldsymbol{x}_i, \boldsymbol{y}_i^*)\}_{i=1}^n$ denote the pairs of crops and ground-truth semantic label

- crop $\boldsymbol{x}_i$ generated from vanilla sample $\boldsymbol{c}_t$ obeys $\|\boldsymbol{x}_i - \boldsymbol{c}_t\|_2 \leq \varepsilon$
- ground-truth semantic label $\boldsymbol{y}_i^* \in \{\gamma_t\}_{t=1}^{\bar{K}}$ of $\boldsymbol{x}_i$ is decided by its corresponding
- the classes are separated: $|\gamma_i - \gamma_k| \geq \delta, \quad \|\boldsymbol{c}_i - \boldsymbol{c}_k\|_2 \geq 2\varepsilon, \quad (\forall i \neq k)$,
- for each sample $\boldsymbol{c}_i$, at most ρn_i augmentations are assigned to wrong labels, where n_i denotes the crop sample number of $\boldsymbol{c}_i$

Theorem 3 (exact label recovery guarantee, informal).

Under proper assumptions, the discrepancy between the label $\bar{\boldsymbol{y}}^t$ estimated by our refinery and the true label $\boldsymbol{y}^*$ of data $\{\boldsymbol{x}_i\}_{i=1}^n$ is bounded:

estimated label

$$\frac{1}{\sqrt{n}}\|\bar{\boldsymbol{y}}^t - \boldsymbol{y}^*\|_2 \leq \frac{1-\alpha_t}{\sqrt{n}}\|\boldsymbol{y} - \boldsymbol{y}^*\|_2 + \alpha_t(6\rho + \zeta),$$

true soft label

where $\zeta = c_6 \varepsilon K^2 \Gamma^5 \xi_3^3 \sqrt{\log K}/\lambda^2(\boldsymbol{C})$, $\boldsymbol{y}^* = [\boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_n^*]$

Performance Guarantee: Exact label recovery

Theorem 3 (exact label recovery guarantee, informal).

Under proper assumptions, the discrepancy between the label $\bar{\boldsymbol{y}}^t$ estimated by our refinery and the true label $\boldsymbol{y}^*$ of data $\{\boldsymbol{x}_i\}_{i=1}^n$ is bounded:

estimated label

$$\frac{1}{\sqrt{n}}\|\bar{\boldsymbol{y}}^t - \boldsymbol{y}^*\|_2 \leq \frac{1-\alpha_t}{\sqrt{n}}\|\boldsymbol{y} - \boldsymbol{y}^*\|_2 + \alpha_t(6\rho + \zeta),$$

true soft label

where $\zeta = c_6 \varepsilon K^2 \Gamma^5 \xi_3^3 \sqrt{\log K}/\lambda^2(\boldsymbol{C})$, $\boldsymbol{y}^* = [\boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_n^*]$

If $\rho \leq \frac{\delta}{24}$, $(1-\alpha_0)|\boldsymbol{y}_i - \boldsymbol{y}_i^*| + \frac{1}{3}\alpha_0\delta < \frac{1}{2}\delta$, the estimated label $\bar{\boldsymbol{y}}_i^t$ predicts true label of $\boldsymbol{y}_i^*$ any crop $\boldsymbol{x}_i$

Exact label recovery: $\gamma_{k^*} = \boldsymbol{y}_i^* \quad \text{with} \quad k^* = \text{argmin}_{1 \leq k \leq \bar{K}} |\bar{\boldsymbol{y}}_i^t - \gamma_k|.$

Performance Guarantee: Exact label recovery

Theorem 3 (exact prediction of network when using label refinery, informal)

Under proper assumptions, by using the refined label $\bar{\boldsymbol{y}}^t$ to train network, the error of network prediction on $\{\boldsymbol{x}_i\}_{i=1}^n$ is upper bounded

predicted label

$$\frac{1}{\sqrt{n}}\|f(\boldsymbol{W}_t, \boldsymbol{X}) - \boldsymbol{y}^*\|_2 \leq 6\rho + \frac{\zeta\lambda(\boldsymbol{C})}{K\Gamma^2\xi_3^2},$$

true label

where $\zeta = c_6\varepsilon K^2\Gamma^5\xi_3^3\sqrt{\log K}/\lambda^2(\boldsymbol{C})$, $\boldsymbol{y}^* = [\boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_n^*]$

If $\rho \leq \frac{\delta}{24}$, $(1-\alpha_0)|\boldsymbol{y}_i - \boldsymbol{y}_i^*| + \frac{1}{3}\alpha_0\delta < \frac{1}{2}\delta$, for any vanilla sample $\boldsymbol{c}_k$, network $f(\boldsymbol{W}_t, \cdot)$ predicts the true semantic label $\boldsymbol{y}_i^*$ of any augmentation $\boldsymbol{x}$ that obeys $\|\boldsymbol{x} - \boldsymbol{c}_k\|_2 \leq \varepsilon$:

Exact label prediction: $\gamma_{k^*} = \gamma_k \quad \text{with} \quad k^* = \text{argmin}_{1\leq i\leq \bar{K}} |f(\boldsymbol{W}_t, \boldsymbol{x}) - \gamma_i|.$

Solution: Self-Labeling Refinement

Self-Labeling Refinement : (1) self-label refinery; (2) momentum mixup

Momentum mixup constructs virtual instance as follows:

$$\boldsymbol{x}_i' = \theta \boldsymbol{x}_i + (1-\theta)\widetilde{\boldsymbol{x}}_k, \quad \boldsymbol{y}_i' = \theta \bar{\boldsymbol{y}}_i + (1-\theta)\bar{\boldsymbol{y}}_k, \tag{1}$$

where $\widetilde{\boldsymbol{x}}_k$ is randomly sampled from the key set $\{\widetilde{\boldsymbol{x}}_i\}_{i=1}^{s}$, $\bar{\boldsymbol{y}}_i$ denotes the refined label by self-labeling refinery, $\theta \in [0,1]$ obeys the beta distribution

Benefits: the component $\widetilde{\boldsymbol{x}}_k$ in $\boldsymbol{x}_i' = \theta \boldsymbol{x}_i + (1-\theta)\widetilde{\boldsymbol{x}}_k$ directly increases the similarity between the query $\boldsymbol{x}_i'$ and its positive key $\widetilde{\boldsymbol{x}}_k$ in $\bar{\boldsymbol{B}}$

momentum mixup can improve the accuracy of the label

Outline

Background: what is contrastive learning and why label is not accurate?

Solution: self-labeling refinery and momentum mixup

Experiments: higher classification accuracy

Conclusion

Outline

Motivation: why one-hot label in MoCo is not accurate?

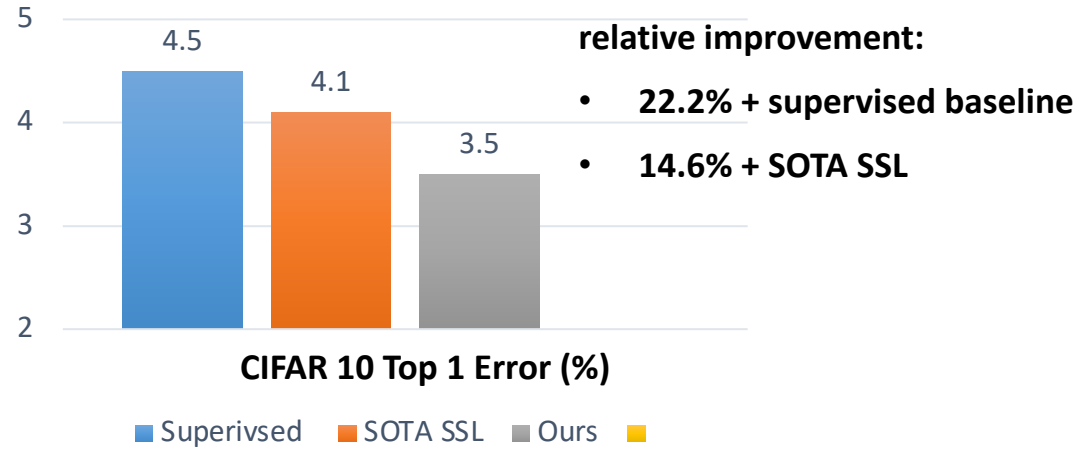
Solution for accurate label: self-labeling refinery and momentum mixup

Experiments: higher classification accuracy

Conclusion

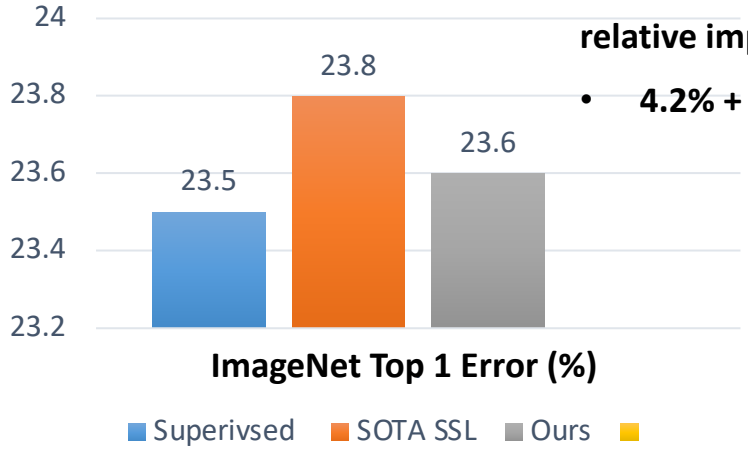
Experimental Results of Proposed NAS Method

CIFAR10 and ImageNet: much lower Top-1 error



relative improvement:

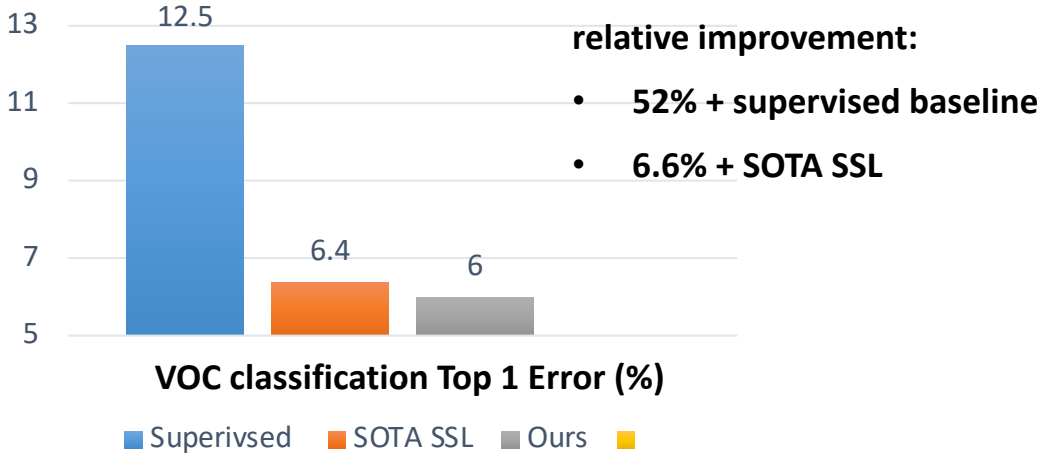
- **22.2% + supervised baseline**
- **14.6% + SOTA SSL**



relative improvement:

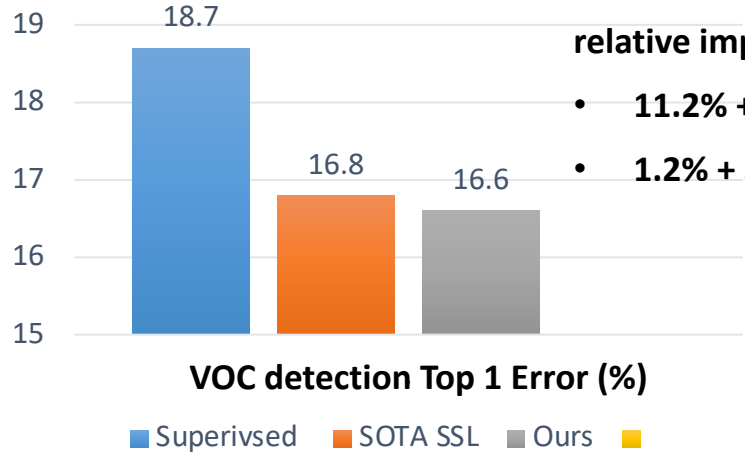
- **4.2% + SOTA SSL**

Downstream tasks: higher performance on VOC classification and detection



relative improvement:

- **52% + supervised baseline**
- **6.6% + SOTA SSL**



relative improvement:

- **11.2% + supervised baseline**
- **1.2% + SOTA SSL**

Outline

Motivation: why one-hot label in MoCo is not accurate?

Solution for accurate label: self-labeling refinery and momentum mixup

Experiments: higher classification accuracy

Conclusion

Conclusion

- **Problems:**

(1) what relationship between one-hot label and the generalization performance?

more precise of labels in contrastive learning, the better the generalization

(2) how to estimate more precise labels?

we propose self-labeling refinery and momentum mixup

Thanks!