Contrastive learning with hard negative samples


Joshua Robinson

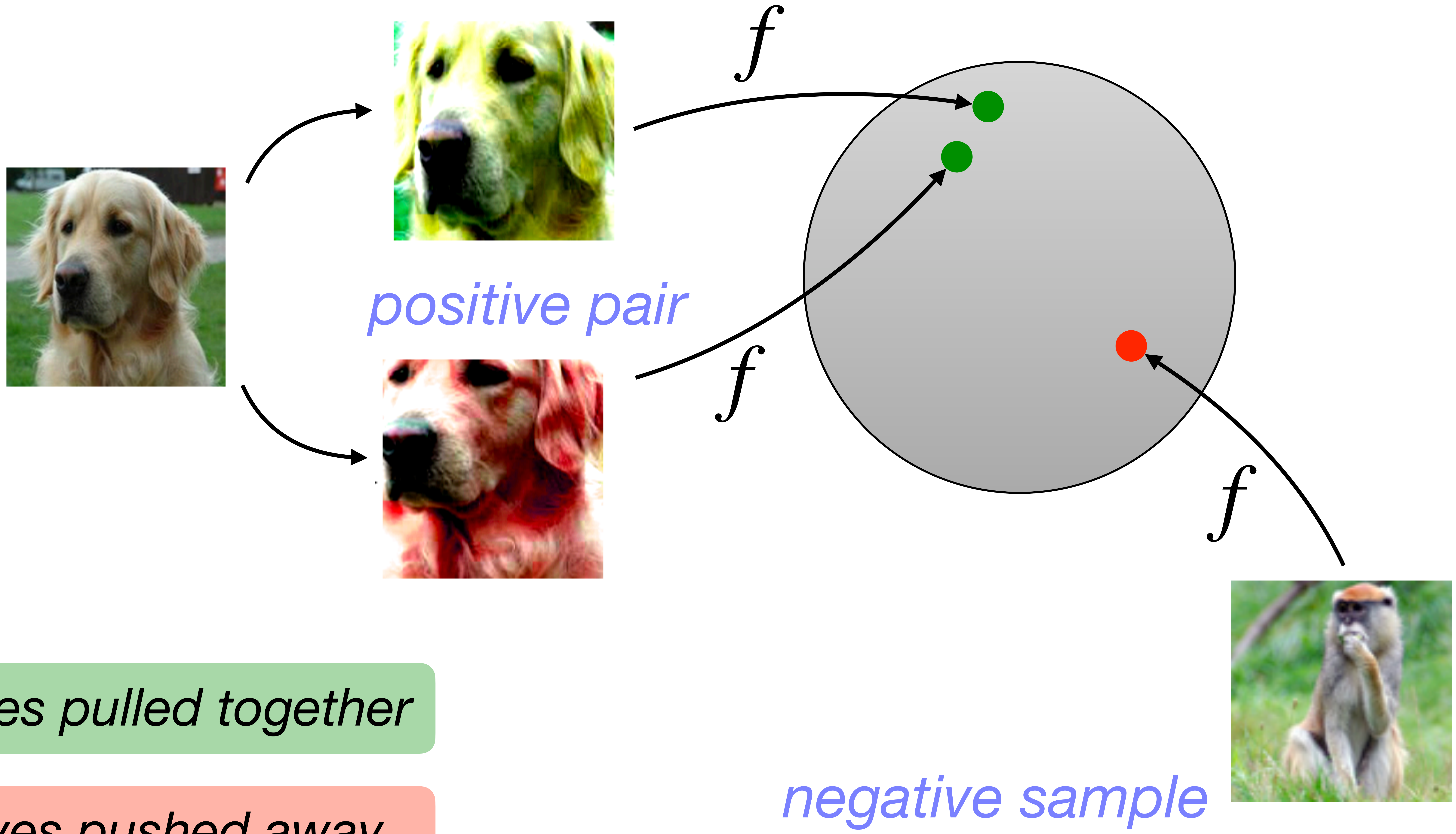

Ching-Yao Chuang


Suvrit Sra


Stefanie Jegelka

Massachusetts Institute of Technology

Contrastive representation learning



positives pulled together

negatives pushed away

How can you generate negative samples?

negatives are typically sampled uniformly at random from training data

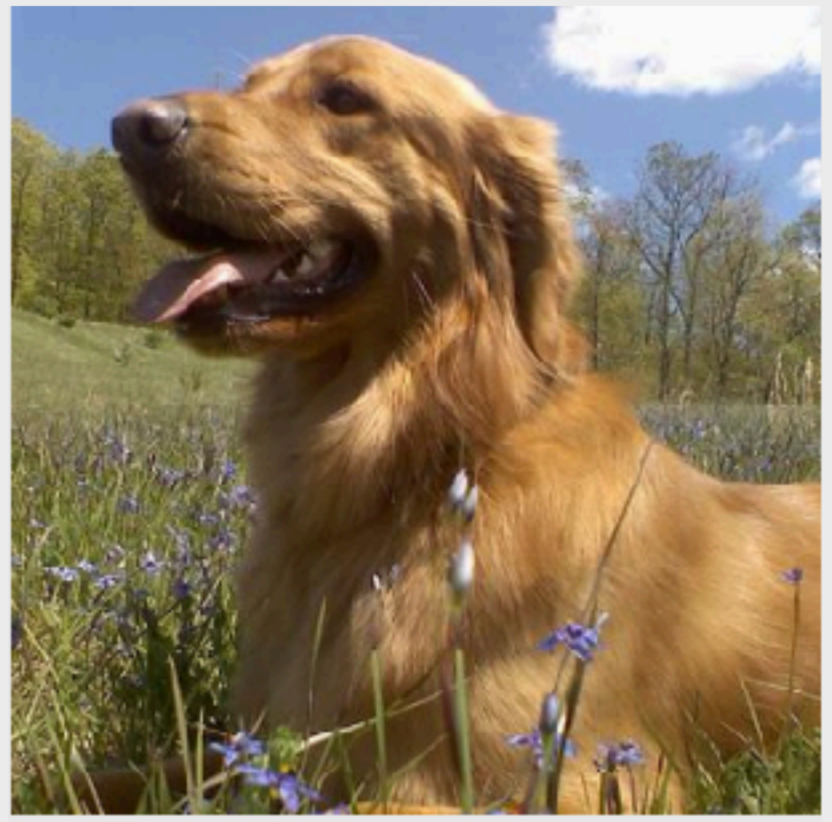
Generating negative samples

Reasons for uniform sampling?

- it is easy to implement
- no supervision to required guide sampling
- large negative batches get good coverage

Negative samples are typically sampled uniformly

What may go wrong with uniform sampling? *Easy negatives*



what if the model already knows they are different?

no useful gradient signal

Hard negative samples

hard negatives are precisely the samples that your encoder is currently "wrong" on

Hard negative sampling

uniform negatives

Sample negatives $\{x_i^-\}_{i=1}^N$ *from marginal* $p(x^-)$

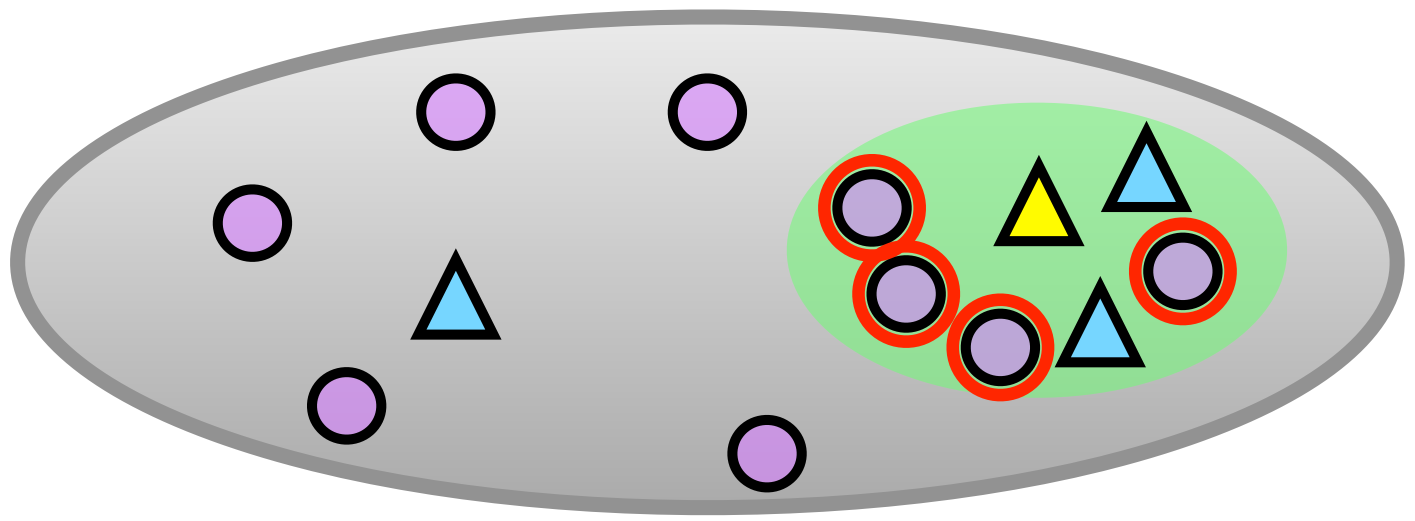
Hard negative sampling

uniform negatives | *Sample negatives* $\{x_i^-\}_{i=1}^N$ *from marginal* $p(x^-)$

hard negatives | *Sample negatives* $\{x_i^-\}_{i=1}^N$ *from*

$$q_\beta(x^-) \propto \boxed{e^{\beta f(x)^\top f(x^-)}} \cdot p(x^-)$$

hard negatives: β controls the level of "hardness"



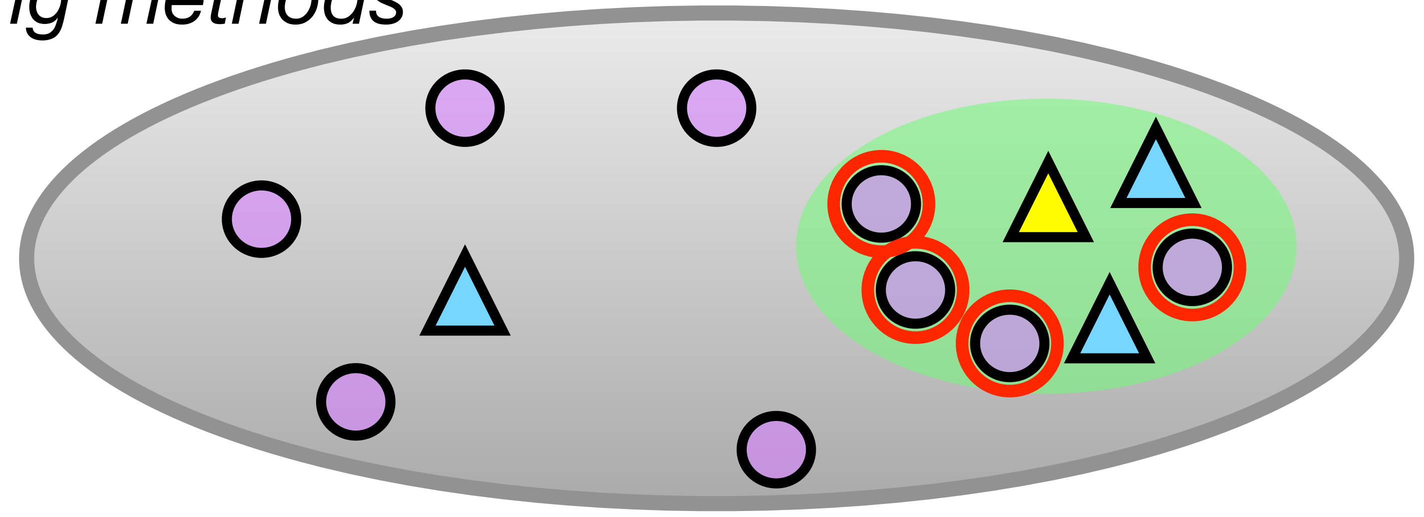
Hard negative sampling

uniform negatives | *Sample negatives* $\{x_i^-\}_{i=1}^N$ *from marginal* $p(x^-)$

hard negatives | *Sample negatives* $\{x_i^-\}_{i=1}^N$ *from* $q_\beta(x^- \,|\, x, x^- \text{diff. class})$ *where* $q_\beta(x^-) \propto e^{\beta f(x)^\top f(x^-)} \cdot p(x^-)$

avoid false hard negatives, approximated using Positive-Unlabeled learning methods

hard negatives: β *controls the level of "hardness"*



Implementation is simple & efficient

our approach implements sampling from q_β using importance sampling using samples from $p(x^-)$ so we can generate in-batch hard negatives

```
# pos      : exp of inner products for positive examples
# neg      : exp of inner products for negative examples
# N        : number of negative examples
# t        : temperature scaling
# tau_plus: class probability
# beta     : concentration parameter

#Original objective
standard_loss = -log(pos.sum() / (pos.sum() + neg.sum()))

#Hard sampling objective (Ours)
reweight = (beta*neg) / neg.mean()
Neg = max((-N*tau_plus*pos + reweight*neg).sum() / (1-tau_plus), e**(-1/t))
hard_loss = -log( pos.sum() / (pos.sum() + Neg))
```

Generalization theory

InfoNCE loss sampling negatives w.r.t q

Theorem (informal):

As $\beta \to \infty$ loss coverages to $\mathscr{L}_\infty(f) = \max_q \mathscr{L}(f; q)$.

Theorem (informal):

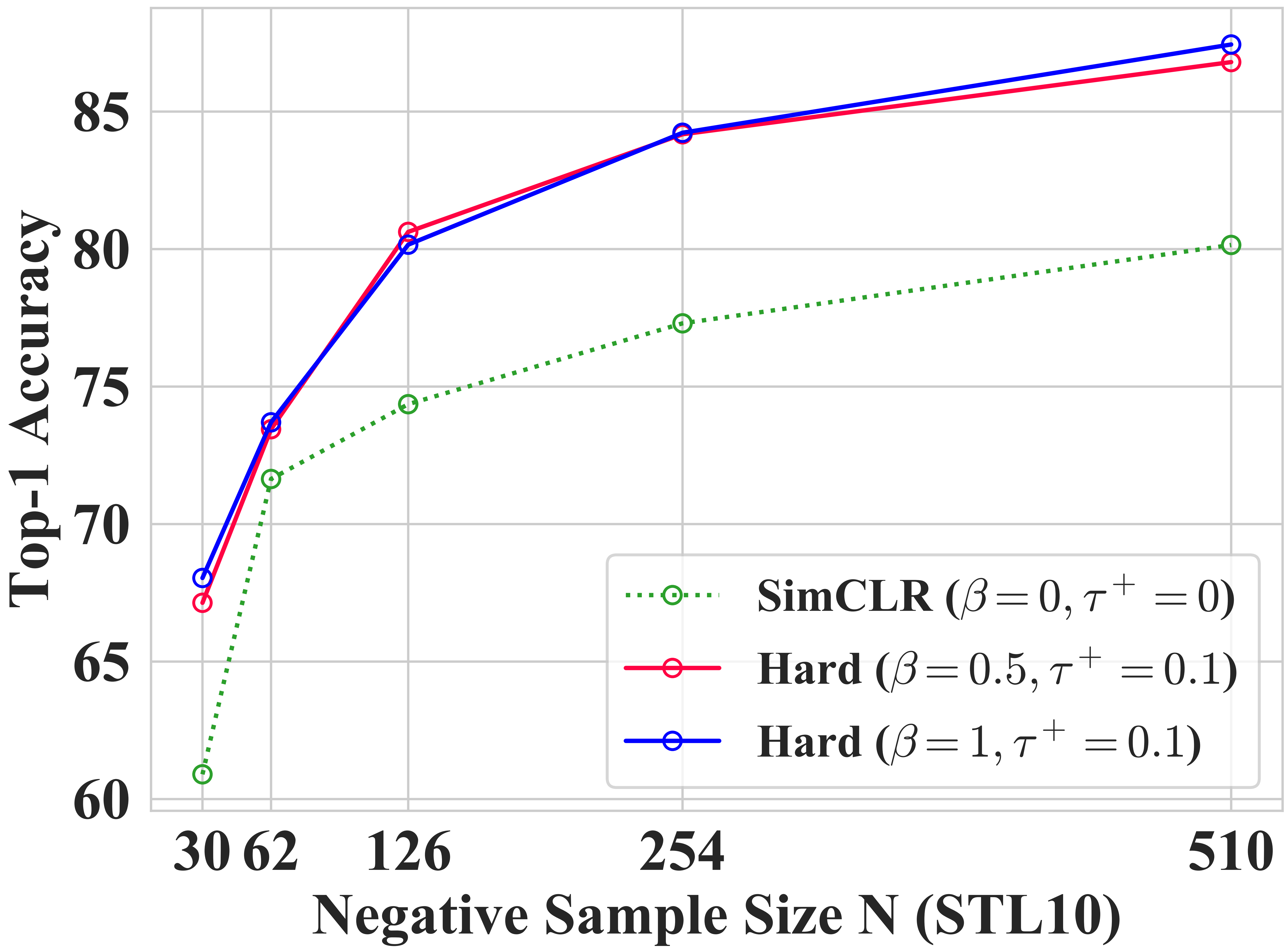
Let be such that f^* of $\mathscr{L}_\infty(f^*) - \inf_f \mathscr{L}_\infty(f) \leq \varepsilon$.

There exists a 1-nearest neighbor classifier h in feature space with classification error $\mathscr{O}(\varepsilon)$

adopting the data generation assumptions proposed in: "A theoretical analysis of contrastive unsupervised representation learning" Saunshi et al. 2019

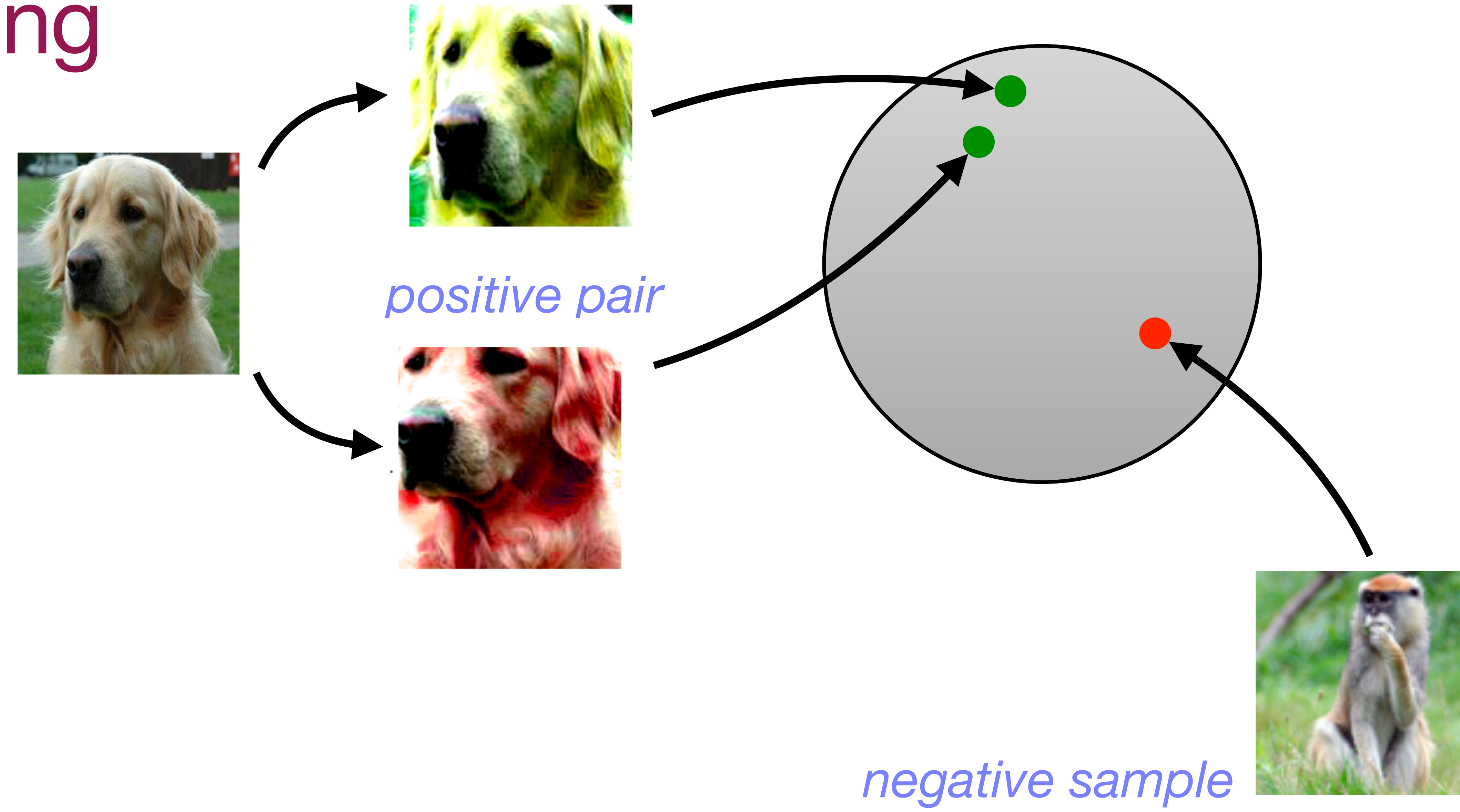
Comparison on vision problems

Linear readout



Summary: negative sampling in contrastive learning

code available



- **not all negatives are created equal:** harder negatives give better learning signal
- **propose hard negative distribution:** prefers x^- with bigger similarity $f(x)^\top f(x^-)$ with anchor x
- **a practical method:** propose simple & efficient practical algorithm based on importance sampling
- **theory:** generalization guarantees for our hard negative sampling method
- **experiments:** observe benefits on multiple vision, natural language, and graph representation problems