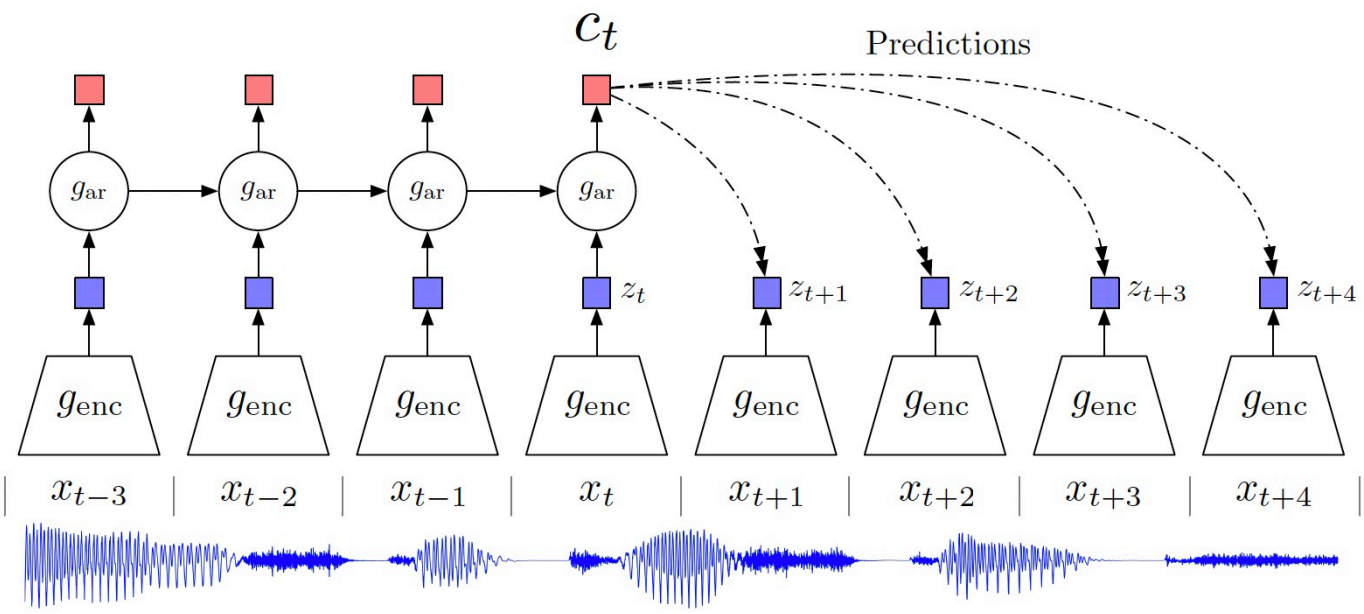
Visual Self-supervised Learning-Contrastive Learning

With negative examples (Contrastive method)

InstDisc:Unsupervised Feature Learning via Non-Parametric Instance Discrimination

- **Motivation:**learn a good feature representation that captures apparent similarity among instances, instead of classes,
- **How:**
  - memory bank store reprsentation
  - NCE loss + Proximal Regularization
  - $$P(i|\mathbf{v}) = \frac{\exp(\mathbf{v}^T\mathbf{f}_i/\tau)}{Z_i}$$
  - $$Z_i = \sum_{j=1}^{n} \exp\left(\mathbf{v}_j^T\mathbf{f}_i/\tau\right)$$
  - $$J_{NCE}(\boldsymbol{\theta}) = -E_{P_d}\left[\log h(i, \mathbf{v}_i^{(t-1)}) - \lambda\|\mathbf{v}_i^{(t)} - \mathbf{v}_i^{(t-1)}\|_2^2\right] -m\cdot E_{P_n}\left[\log(1 - h(i, \mathbf{v}'^{(t-1)}))\right]. \quad (10)$$
- Adv vs Disadv
  - Memory bank update every epoch

CPC: Representation Learning withContrastive Predictive Coding



- Motivation:
  - learn representations by predicting the future in latent space by using powerful autoregressive models. We use a probabilistic contrastive loss which induces the latent space to capture information that is maximally useful to predict future samples
  - learn the representations that encode the underlying shared information between different parts of the (high-dimensional) signal. At the same time it discards low-level information and noise that is more local.
- How:
  - 最大化context和prediction 的互信息
  - When predicting future information we instead encode the target x (future) and context c (present) into a compact distributed vector representations (via non-linear learned mappings) in a way that maximally preserves the mutual information of the original signals x and c defined as

$$I(x;c) = \sum_{x,c} p(x,c) \log \frac{p(x|c)}{p(x)}.$$

  - we do not predict future observations x_{t+k} directly with a generative model $p_k(x_{t+k}|c_t)$. Instead we model a density ratio which preserves the mutual information between x_{t+k} and c_t
  - contrastive loss: q: predictions with context $f(x_{t+k}|c_t)$, k+: enbedding of ground truth, k-: sample out of context

CMC:Contrastive Multiview Coding

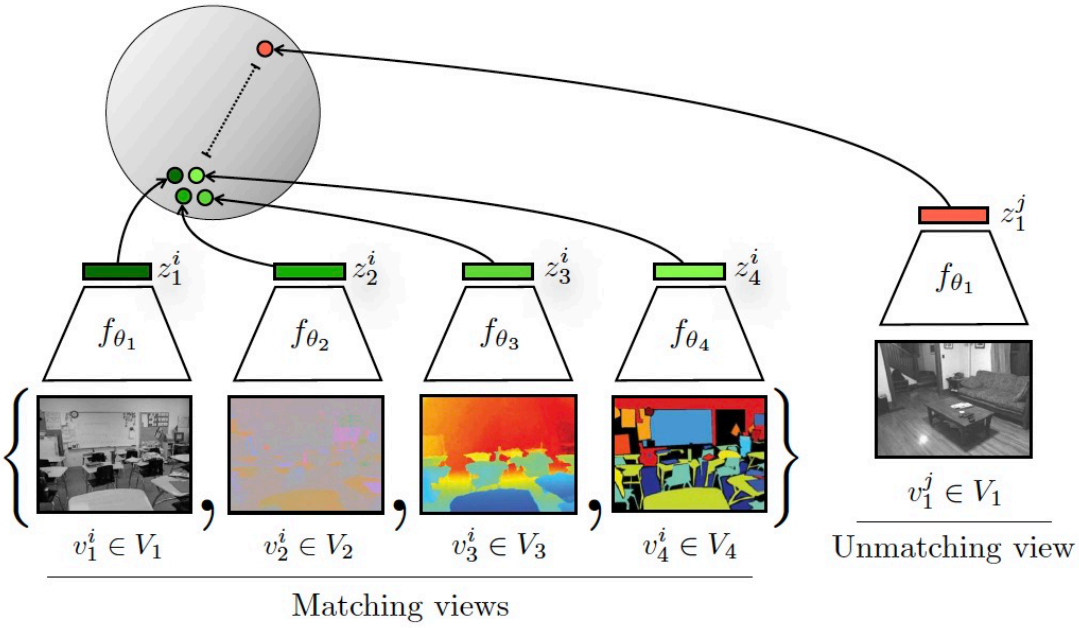


Figure 1: Given a set of sensory views, a deep representation is learnt by bringing views of the *same* scene together in embedding space, while pushing views of *different* scenes apart. Here we show and example of a 4-view dataset (NYU RGBD [53]) and its learned representation. The encodings for each view may be concatenated to form the full representation of a scene.

- Motivation
  - we learn a representation that aims to maximize mutual information between different views of the same scene but is otherwise compact
  - classic hypothesis that a powerful representation is one that models view-invariant factors.
- How: positive:same image of different view,e.g. depth image/segmentation image
- Disadvantage: encoder not shared

MoCo

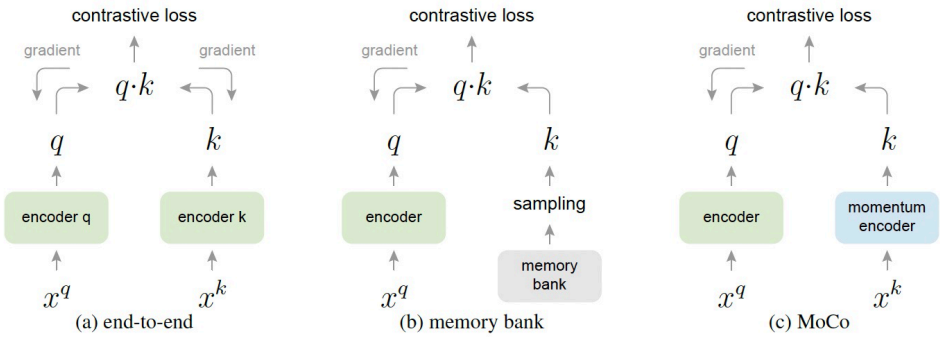


Figure 2. **Conceptual comparison of three contrastive loss mechanisms** (empirical comparisons are in Figure 3 and Table 3). Here we illustrate one pair of query and key. The three mechanisms differ in how the keys are maintained and how the key encoder is updated. **(a)**: The encoders for computing the query and key representations are updated ***end-to-end*** by back-propagation (the two encoders can be different). **(b)**: The key representations are sampled from a ***memory bank*** [61]. **(c)**: ***MoCo*** encodes the new keys on-the-fly by a momentum-updated encoder, and maintains a queue (not illustrated in this figure) of keys.

- we hypothesize that it is desirable to build dictionaries that are: **(i) large and (ii) consistent** as they evolve during training. Intuitively, a larger dictionary may better sample the underlying continuous, highdimensional visual space, while the keys in the dictionary should be represented by the same or similar encoder so that their comparisons to the query are consistent.
- Shuffling BN： The model appears to “cheat” the pretext task and easily finds a low-loss solution. This is possibly because the intra-batch communication among samples (caused by BN) leaks information.
-

SimCLR

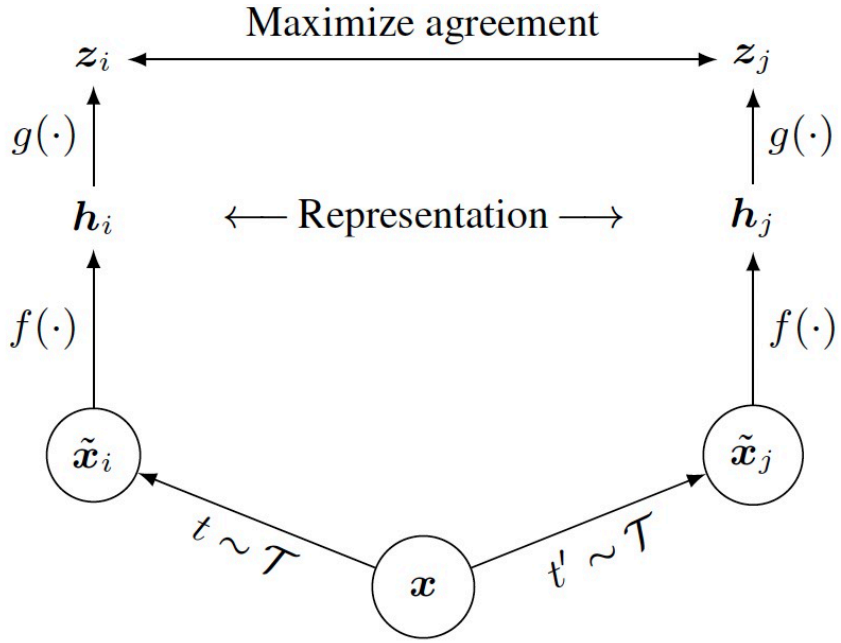


Figure 2. A simple framework for contrastive learning of visual representations. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlated views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training is completed, we throw away the projection head $g(\cdot)$ and use encoder $f(\cdot)$ and representation $\boldsymbol{h}$ for downstream tasks.

- In-batch negative
- composition of data augmentations
  - spatial/geometric transformation of data: cropping and resizing (with horizontal flipping), rotation
  - appearance transformation, such as color distortion (including color dropping, brightness, contrast, saturation, hue)
  - it is critical to *compose cropping with color distortion** in order to learn generalizable features.
- global BN:
- Large batch size:
- nonlinear projection head:

We conjecture that the importance of **using the representation before the nonlinear projection is due to loss of information induced by the contrastive loss**. **In particular, $z = g(h)$ is trained to be invariant to data transformation. Thus, g can remove information that may be useful for the downstream task, such as the color or orientation of objects**. By leveraging the nonlinear transformation $g(\cdot)$, more information can be formed and maintained in h.

- **L2 normalization** (i.e. cosine similarity) along with temperature effectively weights different examples, and an **appropriate temperature** can help the model learn from **hard negatives**; and 2) unlike cross-entropy, other objective functions do not weigh the negatives by their relative hardness.

Name	Negative loss function	Gradient w.r.t. $\boldsymbol{u}$
NT-Xent	$\boldsymbol{u}^T\boldsymbol{v}^+/\tau - \log \sum_{\boldsymbol{v}\in\{\boldsymbol{v}^+,\boldsymbol{v}^-\}} \exp(\boldsymbol{u}^T\boldsymbol{v}/\tau)$	$(1 - \frac{\exp(\boldsymbol{u}^T\boldsymbol{v}^+/\tau)}{Z(\boldsymbol{u})})/\tau\boldsymbol{v}^+ - \sum_{\boldsymbol{v}^-} \frac{\exp(\boldsymbol{u}^T\boldsymbol{v}^-/\tau)}{Z(\boldsymbol{u})}/\tau\boldsymbol{v}^-$
NT-Logistic	$\log \sigma(\boldsymbol{u}^T\boldsymbol{v}^+/\tau) + \log \sigma(-\boldsymbol{u}^T\boldsymbol{v}^-/\tau)$	$(\sigma(-\boldsymbol{u}^T\boldsymbol{v}^+/\tau))/\tau\boldsymbol{v}^+ - \sigma(\boldsymbol{u}^T\boldsymbol{v}^-/\tau)/\tau\boldsymbol{v}^-$
Margin Triplet	$-\max(\boldsymbol{u}^T\boldsymbol{v}^- - \boldsymbol{u}^T\boldsymbol{v}^+ + m, 0)$	$\boldsymbol{v}^+ - \boldsymbol{v}^-$ if $\boldsymbol{u}^T\boldsymbol{v}^+ - \boldsymbol{u}^T\boldsymbol{v}^- < m$ else $\mathbf{0}$

Table 2. Negative loss functions and their gradients. All input vectors, i.e. $\boldsymbol{u}, \boldsymbol{v}^+, \boldsymbol{v}^-$, are ℓ_2 normalized. NT-Xent is an abbreviation for "Normalized Temperature-scaled Cross Entropy". Different loss functions impose different weightings of positive and negative examples.

SwAV:Unsupervised Learning of Visual Features by Contrasting Cluster Assignments

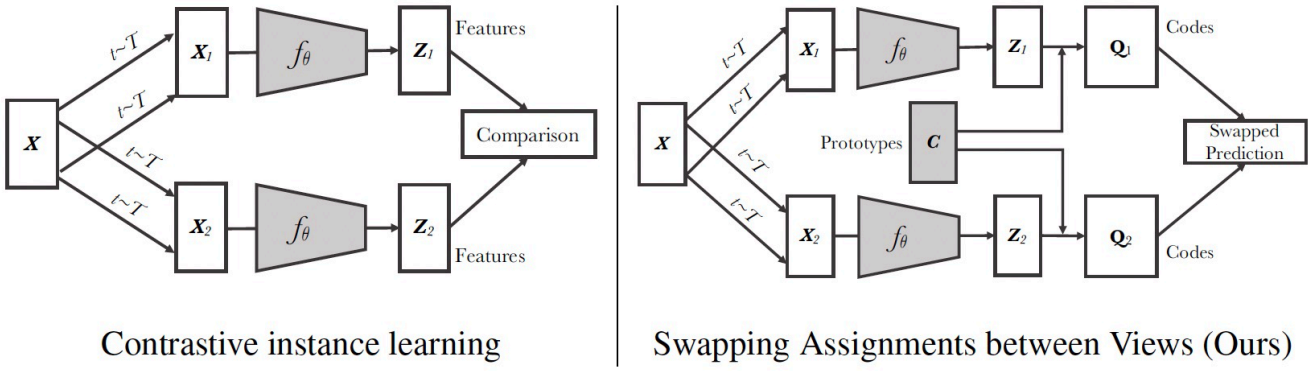


Figure 1: **Contrastive instance learning (left) *vs*. SwAV (right).** In contrastive learning methods applied to instance classification, the features from different transformations of the same images are compared directly to each other. In SwAV, we first obtain "codes" by assigning features to prototype vectors. We then solve a "swapped" prediction problem wherein the codes obtained from one data augmented view are predicted using the other view. Thus, SwAV does not directly compare image features. Prototype vectors are learned along with the ConvNet parameters by backpropragation.

- Motivation:
  - SwAV, that takes advantage of contrastive methods without requiring to compute pairwise comparisons. Specifically, our method simultaneously clusters the data while **enforcing consistency between cluster assignments produced for different augmentations (or "views") of the same image**, instead of comparing features directly as in contrastive learning.
- How

$$\ell(\mathbf{z}_t, \mathbf{q}_s) = -\sum_k \mathbf{q}_s^{(k)} \log \mathbf{p}_t^{(k)}, \quad \text{where} \quad \mathbf{p}_t^{(k)} = \frac{\exp\left(\frac{1}{\tau}\mathbf{z}_t^\top \mathbf{c}_k\right)}{\sum_{k'} \exp\left(\frac{1}{\tau}\mathbf{z}_t^\top \mathbf{c}_{k'}\right)}.$$

  - Multi-Crop
  -

w/o negative examples

BYOL

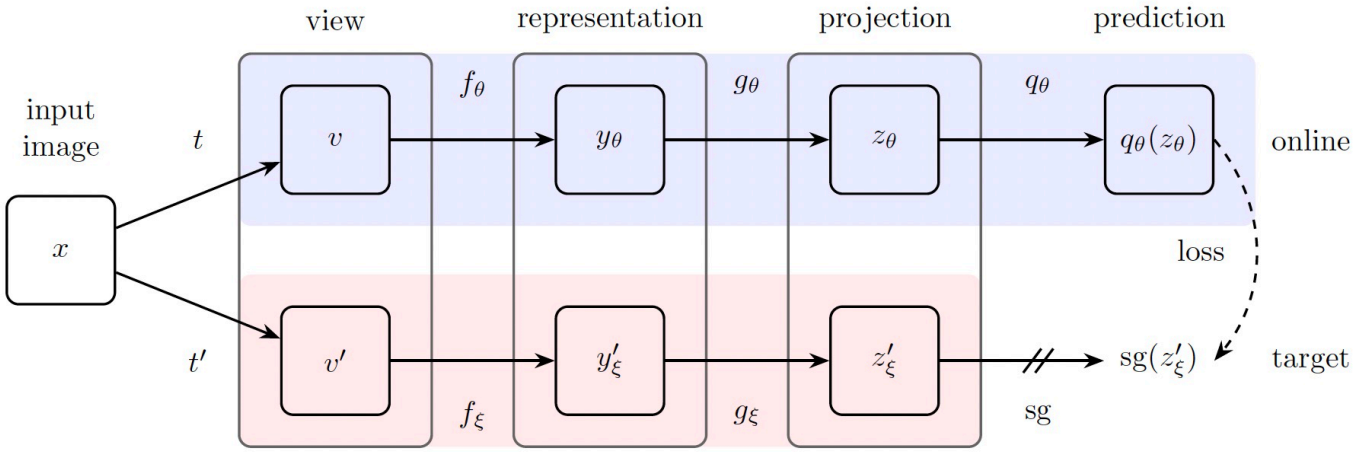


Figure 2: BYOL's architecture. BYOL minimizes a similarity loss between $q_\theta(z_\theta)$ and $\mathrm{sg}(z'_\xi)$, where θ are the trained weights, ξ are an exponential moving average of θ and sg means stop-gradient. At the end of training, everything but f_θ is discarded, and y_θ is used as the image representation.

$$\mathcal{L}_{\theta,\xi} \triangleq \left\| \overline{q_\theta}(z_\theta) - \overline{z}'_\xi \right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\left\| q_\theta(z_\theta) \right\|_2 \cdot \left\| z'_\xi \right\|_2}.$$

$$\theta \leftarrow \mathrm{optimizer}\left(\theta, \nabla_\theta \mathcal{L}^{\mathtt{BYOL}}_{\theta,\xi}, \eta\right),$$
$$\xi \leftarrow \tau\xi + (1-\tau)\theta,$$

SimSiam:Exploring Simple Siamese Representation Learning

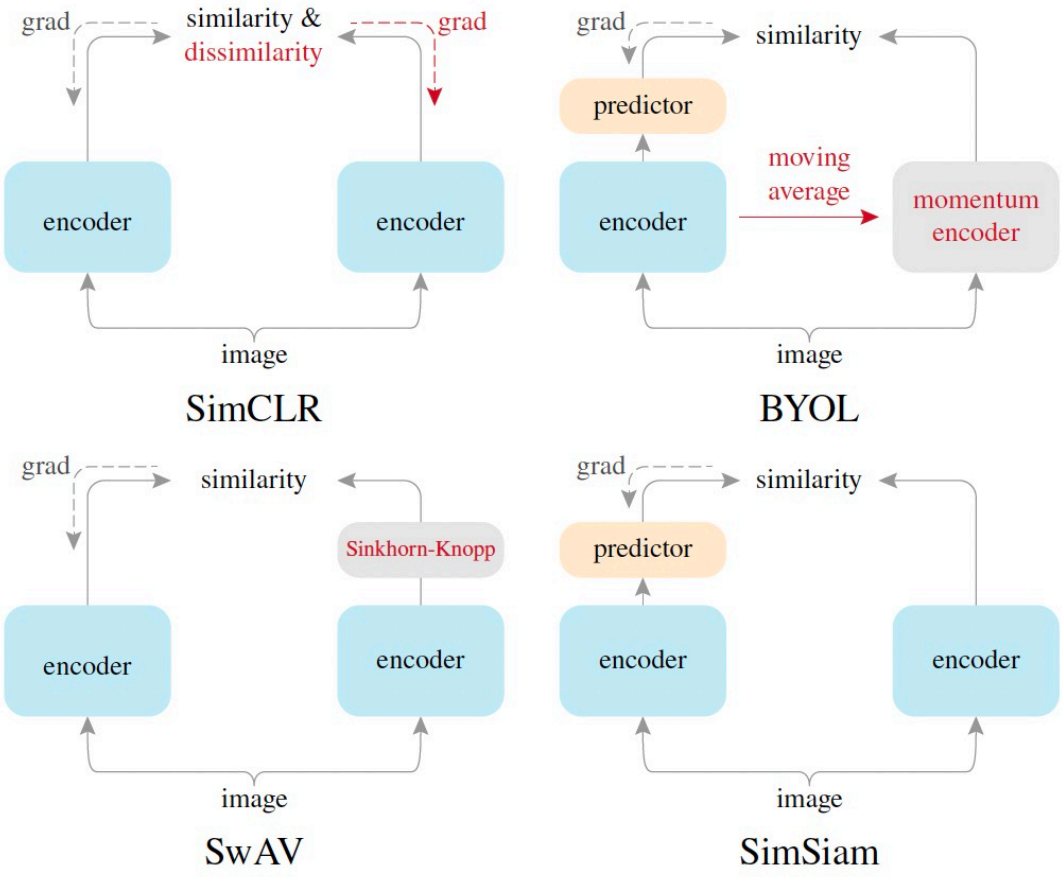


Figure 3. **Comparison on Siamese architectures**. The encoder includes all layers that can be shared between both branches. The dash lines indicate the gradient propagation flow. In BYOL, SwAV, and SimSiam, the lack of a dash line implies stop-gradient, and their symmetrization is not illustrated for simplicity. The components in red are those missing in SimSiam.

- Motivation:that collapsing solutions do exist for the loss and structure, but a **stop-gradient** operation plays an essential role in preventing collapsing.
- SimSiam as EM algorithm

  考虑以下损失函数:

  $$\mathcal{L} = \mathbb{E}_{x,\mathcal{T}} \left[||\mathcal{F}_\theta(\mathcal{T}(x)) - \eta_x||_2^2 \right]$$

  η_x是图片x的表示，是一组可学习的参数，θ是网络参数。我们采用EM算法优化这两组参数。目标是:

  $$\theta^*, \eta^* = \min \mathcal{L}(\theta, \eta)$$

  我们交替优化:

  $$\theta^t \leftarrow \arg\min_\theta \mathcal{L}(\theta, \eta^{t-1})$$

  $$\eta^t \leftarrow \arg\min_\eta \mathcal{L}(\theta^t, \eta)$$

  - E步：优化θ：采用sgd优化θ 需要采用stop gradient防止梯度回传到η，因为η被视为常量
  - M步：优化η_x :给定每个x并且固定$\theta = \theta_t$优化$\mathcal{L}$（求导）：

$$\eta_x^t = \arg\min_{\eta_x} \mathbb{E}_{\mathcal{T}}[||\mathcal{F}_{\theta^t}(\mathcal{T}(x)) - \eta_x||_2^2] = \mathbb{E}_{\mathcal{T}}[\mathcal{F}_{\theta^t}(\mathcal{T}(x))]$$

说明η_x的最优解是x经过augmentation后表征的期望(关于augmentation)

- One-step alternation
  - 通过采样**一次**augmentation $\mathcal{T}'$对η_x^t估计：

$$\eta_x^t = \mathcal{F}_{\theta^t}(\mathcal{T}'(x))$$

  - 对θ优化,带入上式,采用sgd优化

$$\theta^{t+1} = \arg\min \mathbb{E}_{x,\mathcal{T}}[||\mathcal{F}_{\theta}(\mathcal{T}'(x)) - \eta_x^t||]$$

- Predictor

predictor h是对$\mathbb{E}_{\mathcal{T}}[\mathcal{F}_{\theta^t}(\mathcal{T}(x))]$的估计：

$$h^*(z_1) = \arg\min_{h(z_1)} \mathbb{E}_z[||h(z_1) - z_2||_2^2] = \mathbb{E}_z[z_2] = \mathbb{E}_{\mathcal{T}}[f(\mathcal{T}(x))]$$

在单步优化中,$\mathbb{E}_{\mathcal{T}}[\cdot]$被忽略了，$h$视为对其的估计(Expectation over augmentations)，除了用h估计$\mathbb{E}_{\mathcal{T}}[\cdot]$，也可以采用动量更新的方式

- Symmetrization

Actually, the SGD optimizer computes the empirical expectation of $\mathbb{E}_{x,\mathcal{T}}[\cdot]$ by sampling a batch of images and one pair of augmentations $(\mathcal{T}_1, \mathcal{T}_2)$. In principle, the empirical expectation should be more precise with **denser sampling**,Symmetrization supplies an extra pair $(\mathcal{T}_2, \mathcal{T}_2)$ This explains that symmetrization is **not necessary** for our method to work, yet it is able to improve accuracy

DINO

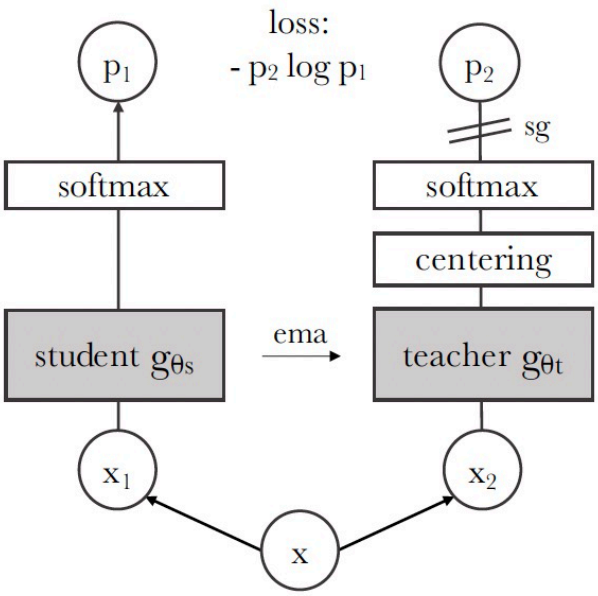


Figure 2: **Self-distillation with no labels.** We illustrate DINO in the case of one single pair of views (x_1, x_2) for simplicity. The model passes two different random transformations of an input image to the student and teacher networks. Both networks have the same architecture but different parameters. The output of the teacher network is centered with a mean computed over the batch. Each networks outputs a K dimensional feature that is normalized with a temperature softmax over the feature dimension. Their similarity is then measured with a cross-entropy loss. We apply a stop-gradient (sg) operator on the teacher to propagate gradients only through the student. The teacher parameters are updated with an exponential moving average (ema) of the student parameters.