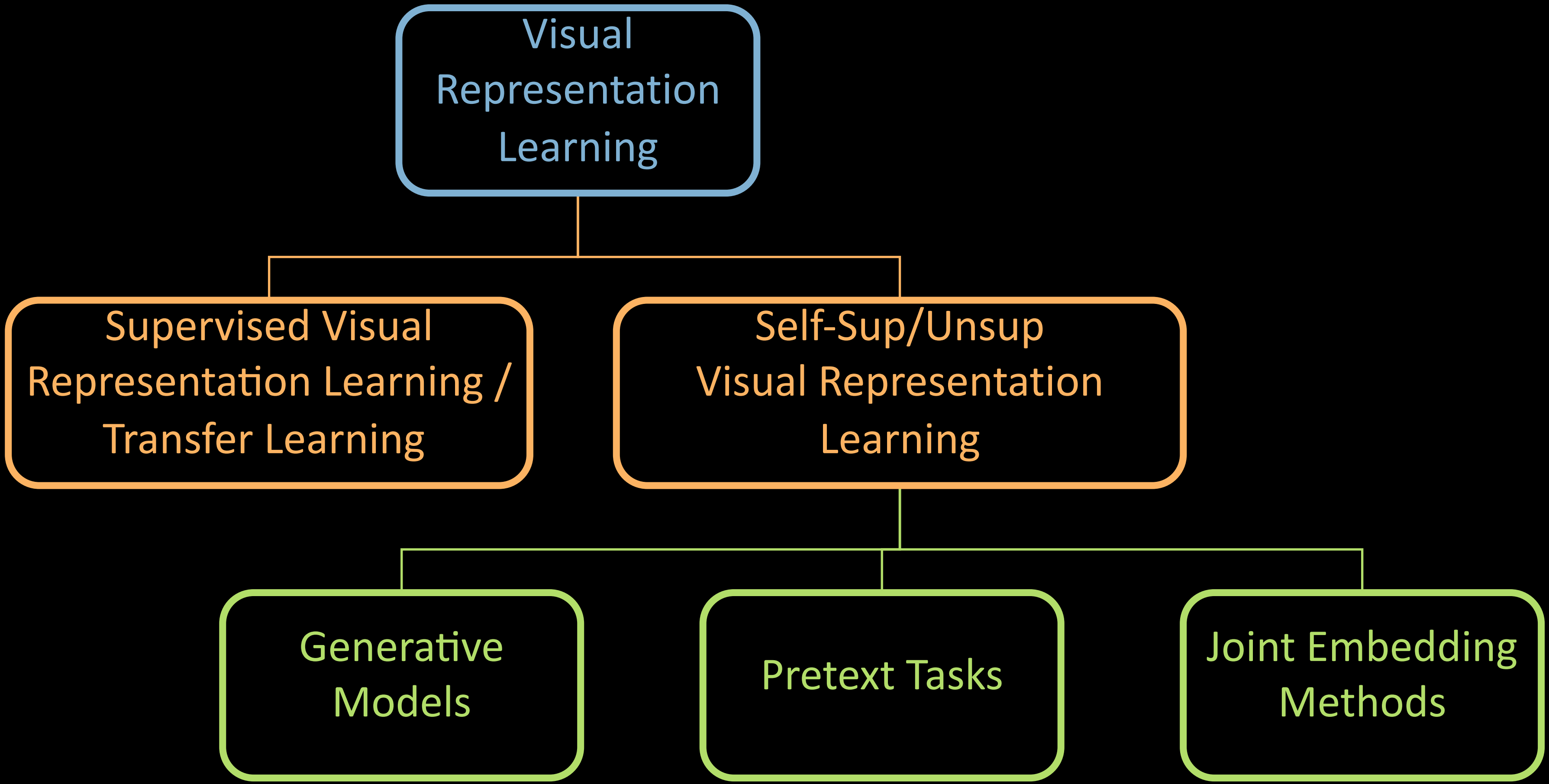
Joint Embedding Methods

Self-Supervised Visual Representation Learning

Visual representation learning

Overview

Visual Representation Learning



Self-supervised visual representation learning

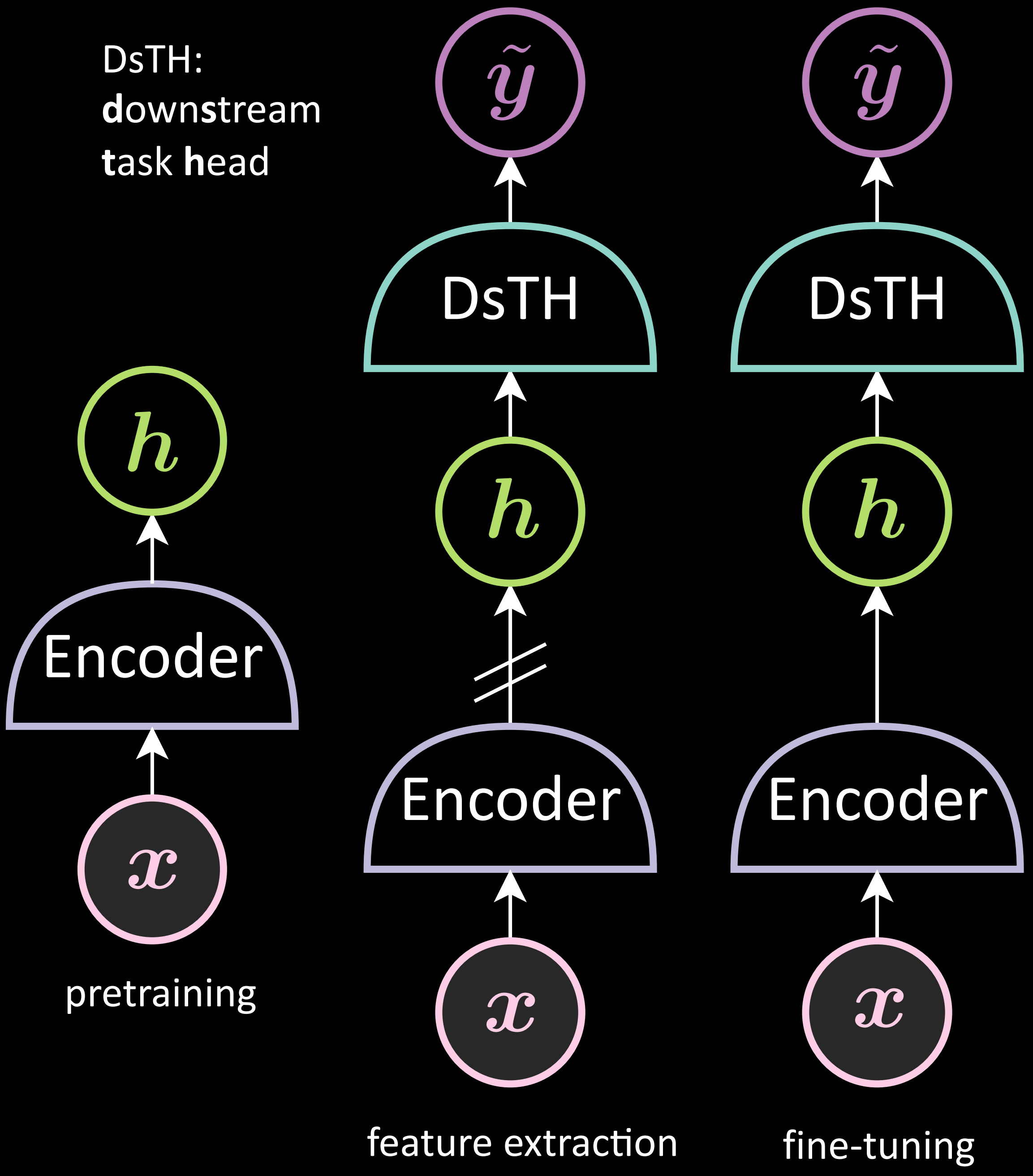
Step 1: pretraining

Use a large amount of unlabeled data to train a backbone network

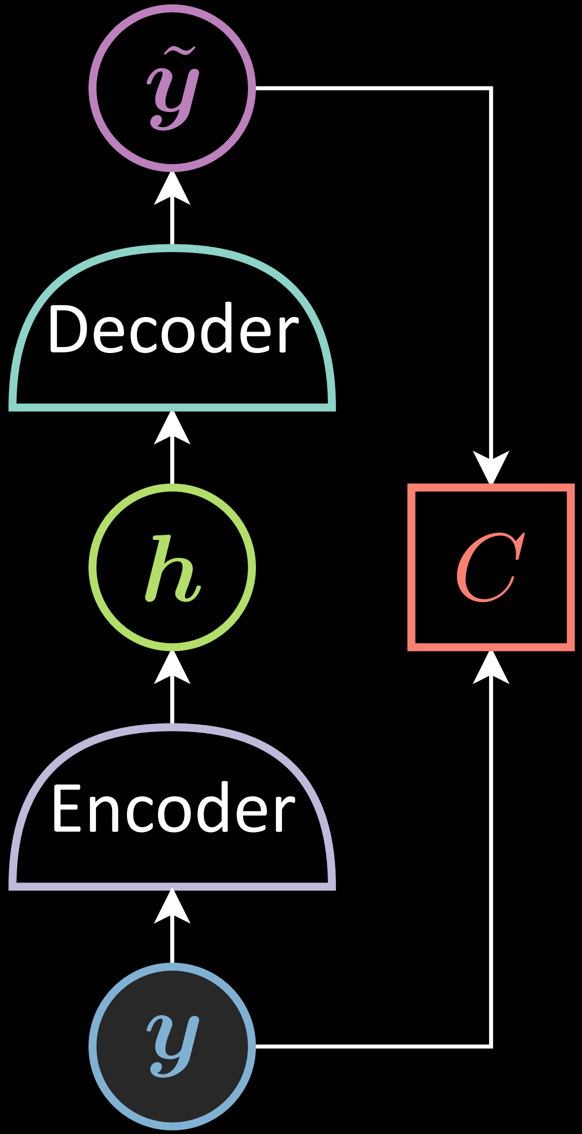
different methods will produce the backbone network differently

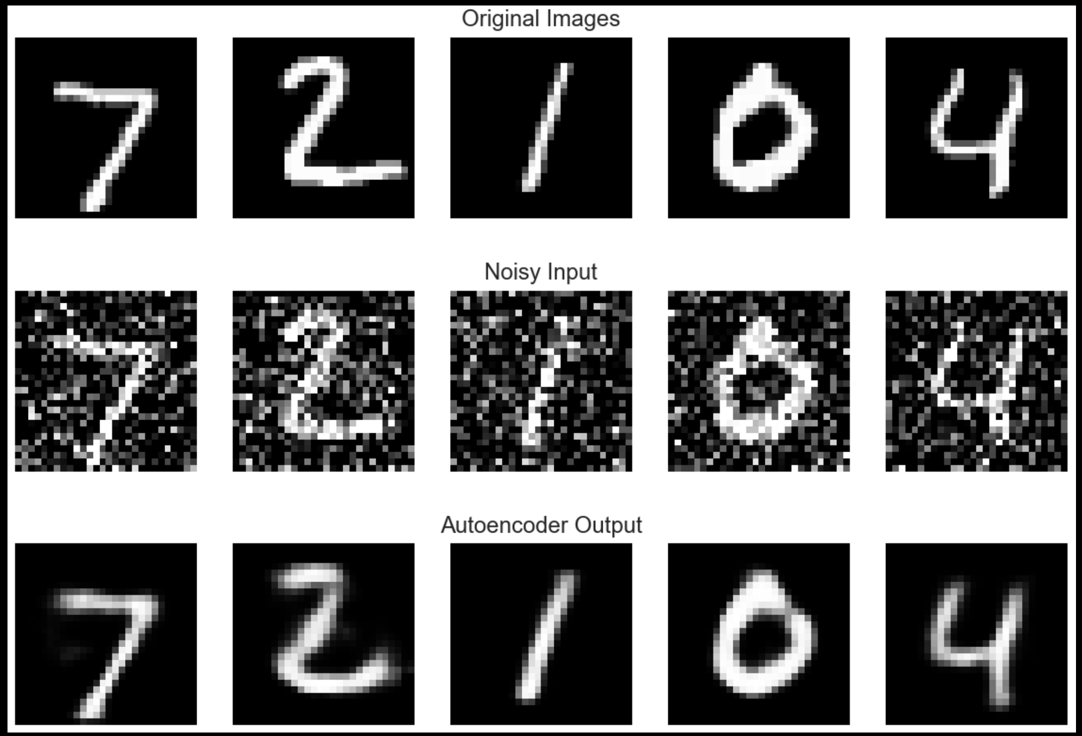
Step 2: evaluation

Use a small amount of labeled data to train a downstream task head network

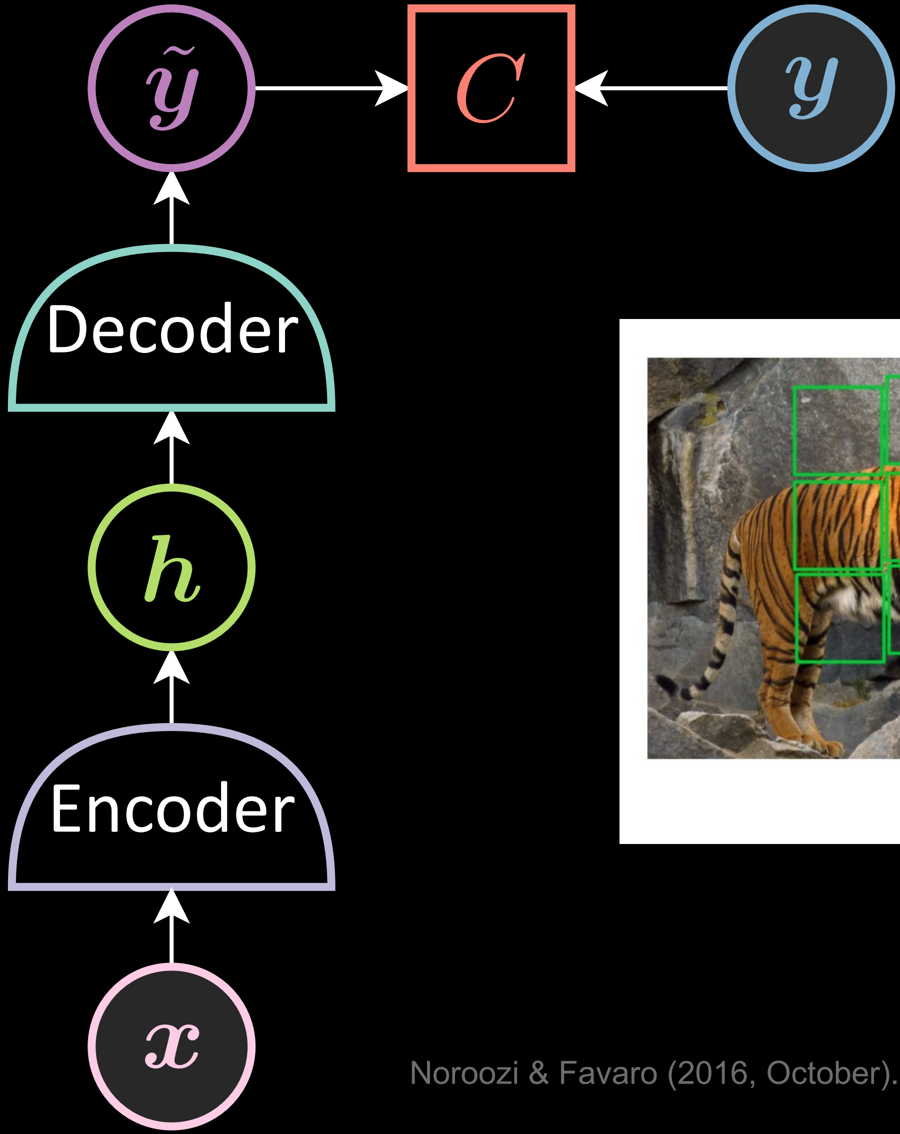


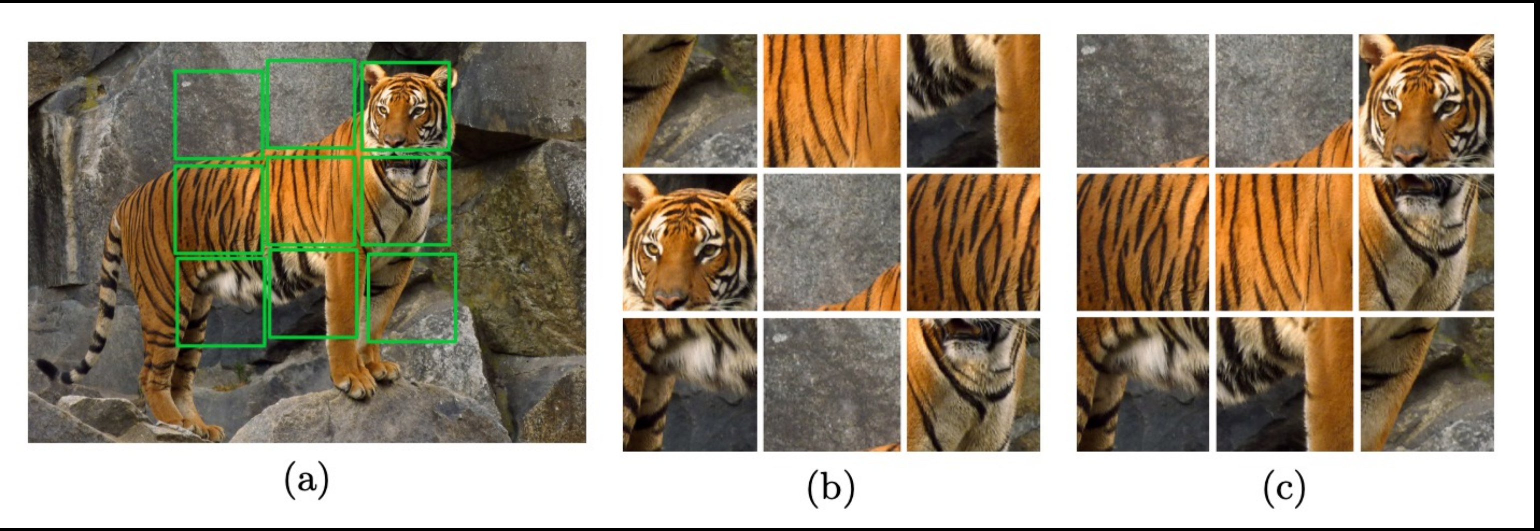
Generative Models- Autoencoder





Pretext Tasks





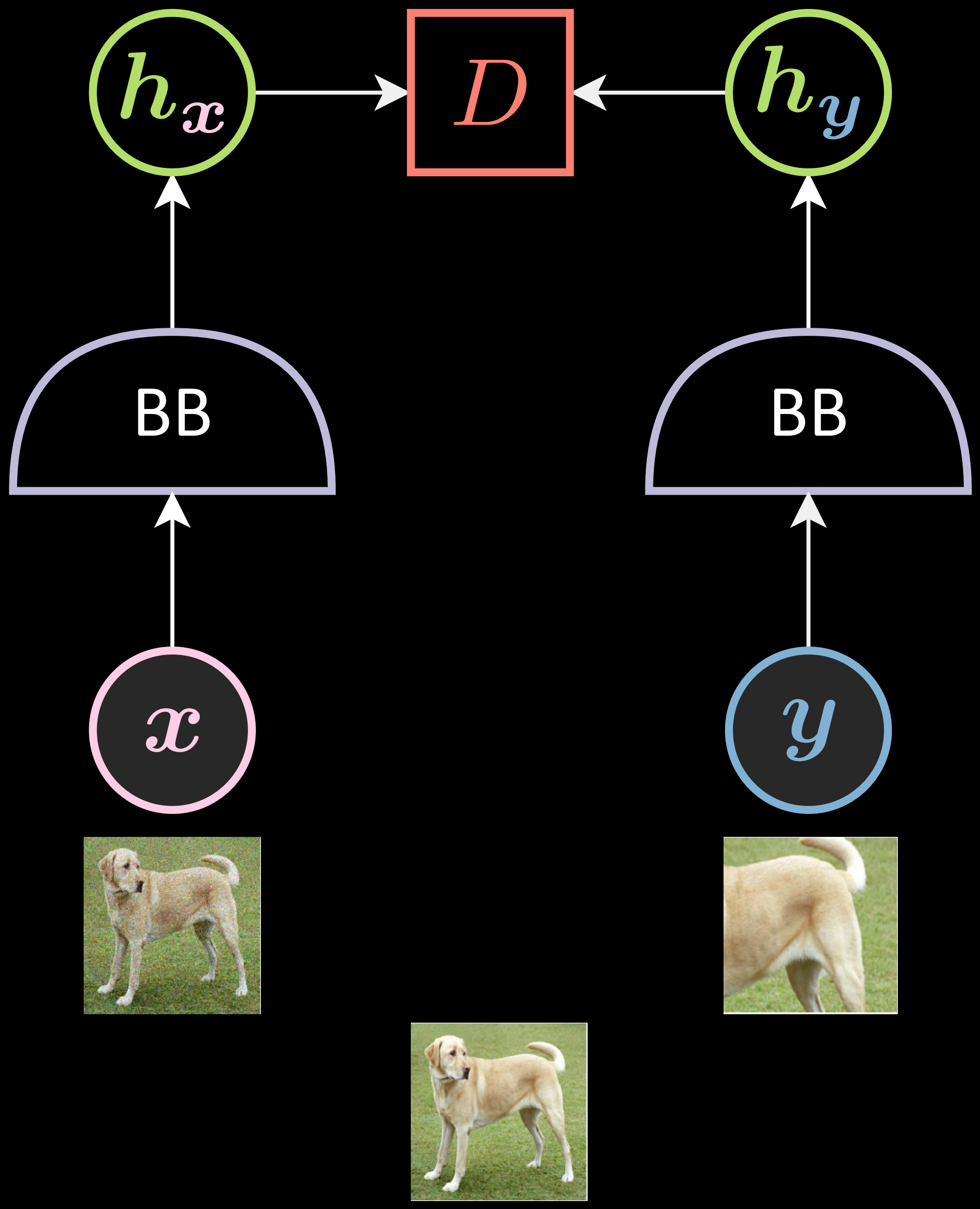
Noroozi & Favaro (2016, October). Unsupervised learning of visual representations by solving jigsaw puzzles.

Joint embedding methods

Siamese nets & co.

Joint Embedding Methods

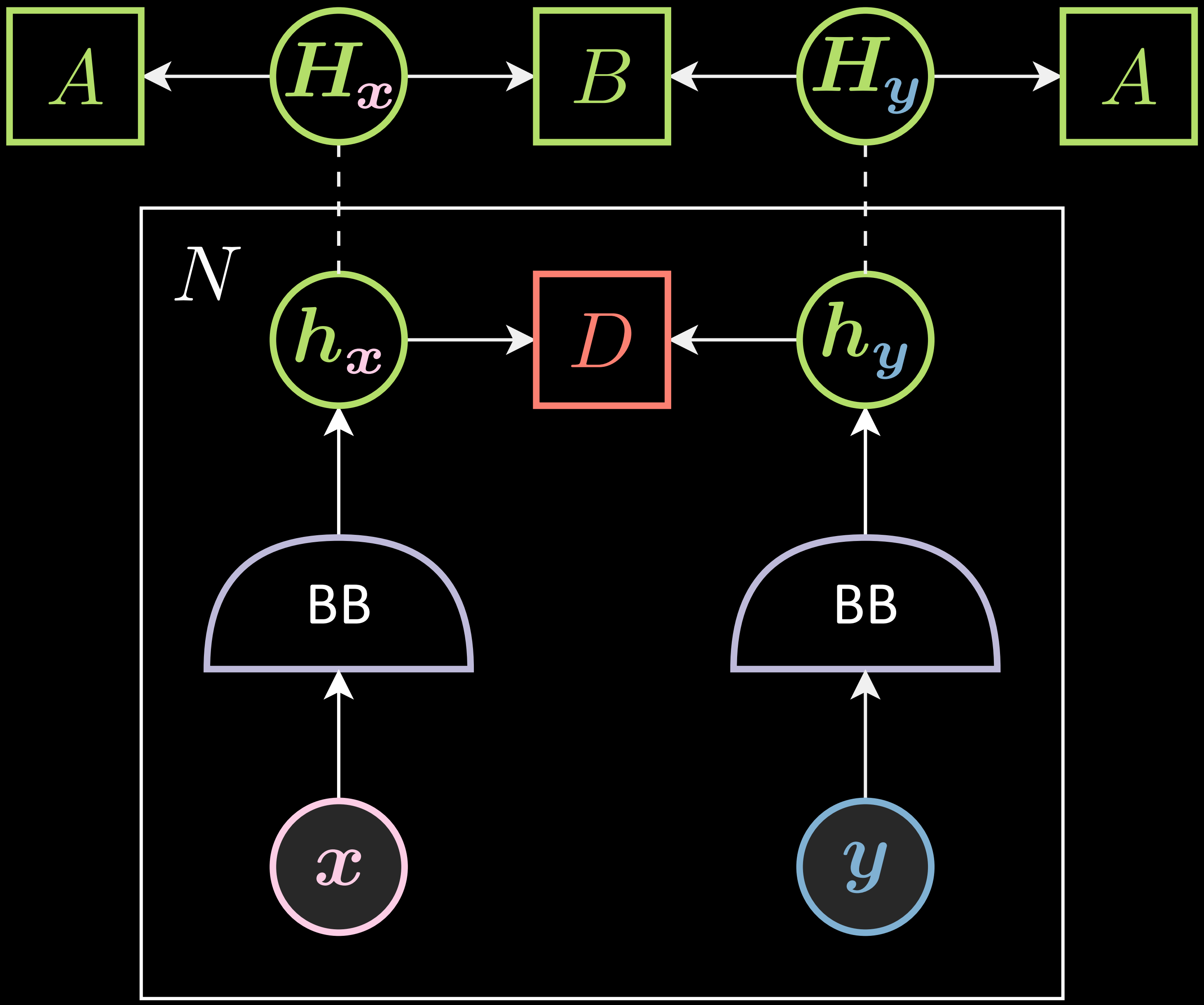
Good backbone network should be robust to certain distortions (invariant to data augmentation)



BB: backbone

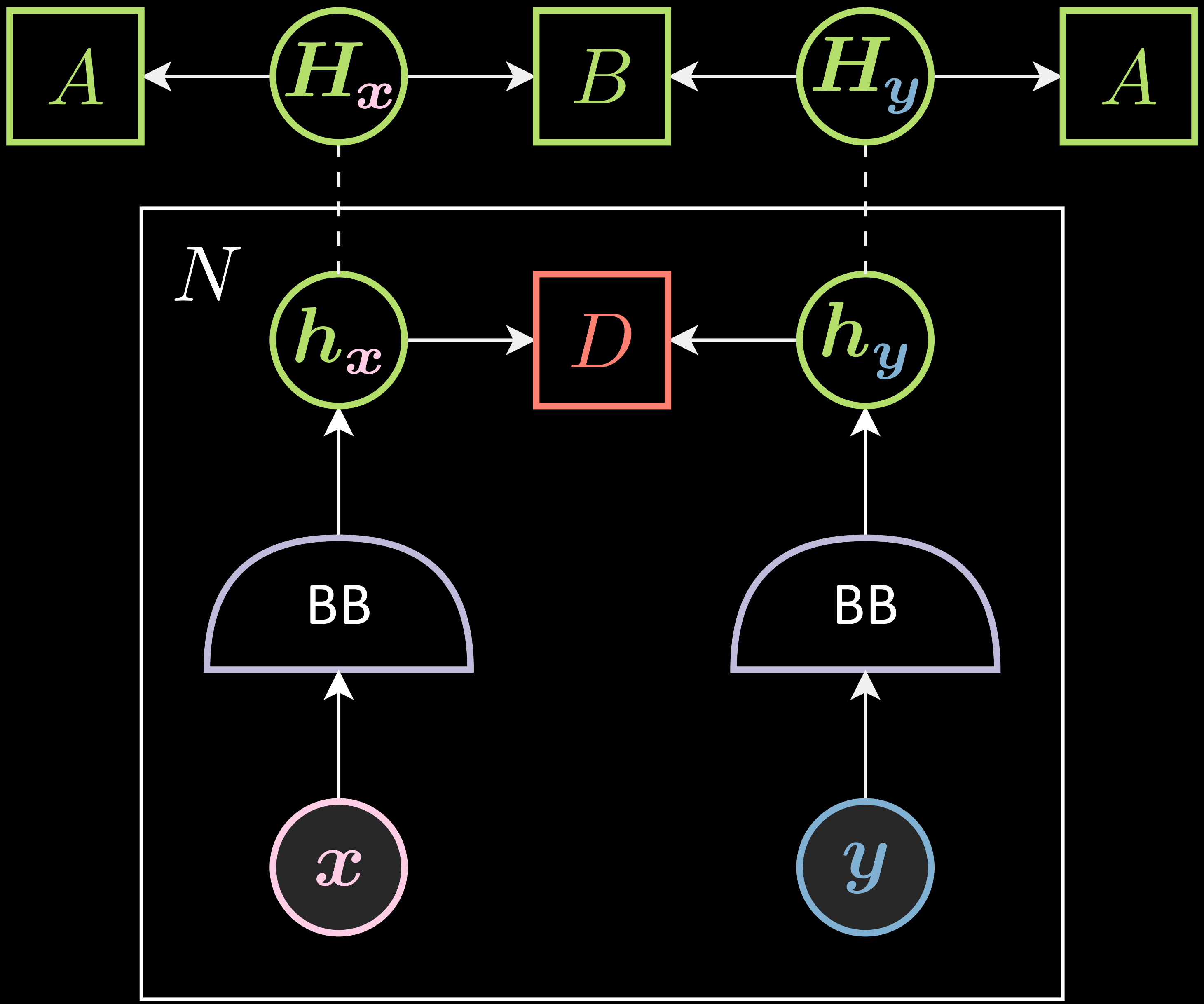
JEM

We add extra loss to prevent the trivial solution (constant embeddings)



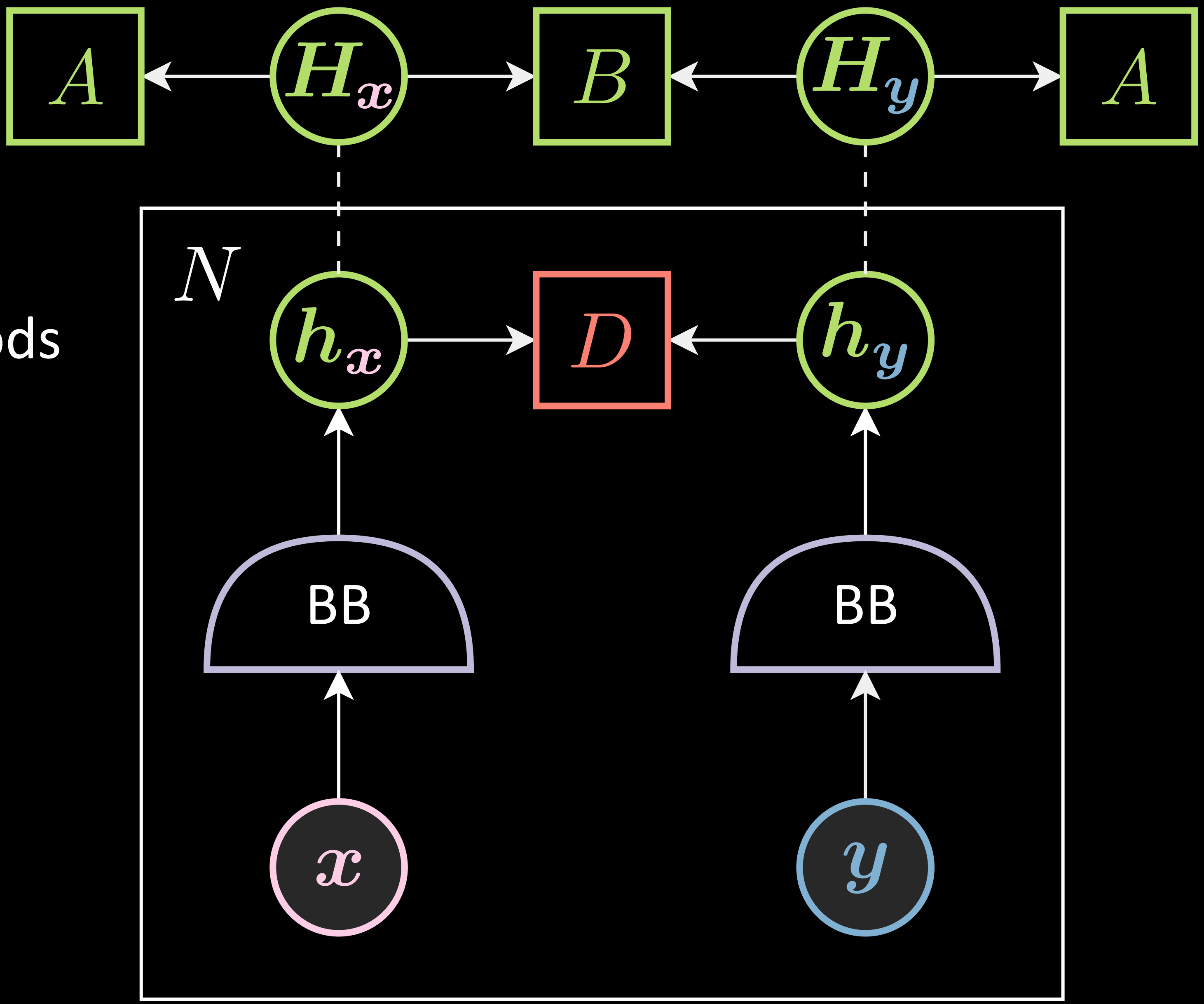
JEM

1. Data augmentation
2. Backbone network
3. Energy function
4. Loss functional



JEM

1. Contrastive methods
2. Non-contrastive methods
3. Clustering methods
4. Other methods



JEM loss functions

- A term that pushes the positive pair closer
- An (implicit) term that prevents the trivial solution (constant output)

To make the training stable, people usually normalize the embeddings or put a hinge on the loss function to prevent the norm of embeddings becoming too large or too small

Contrastive methods

Pull up on contrastive samples

Contrastive Methods

The loss function should push

1. the positive pairs closer

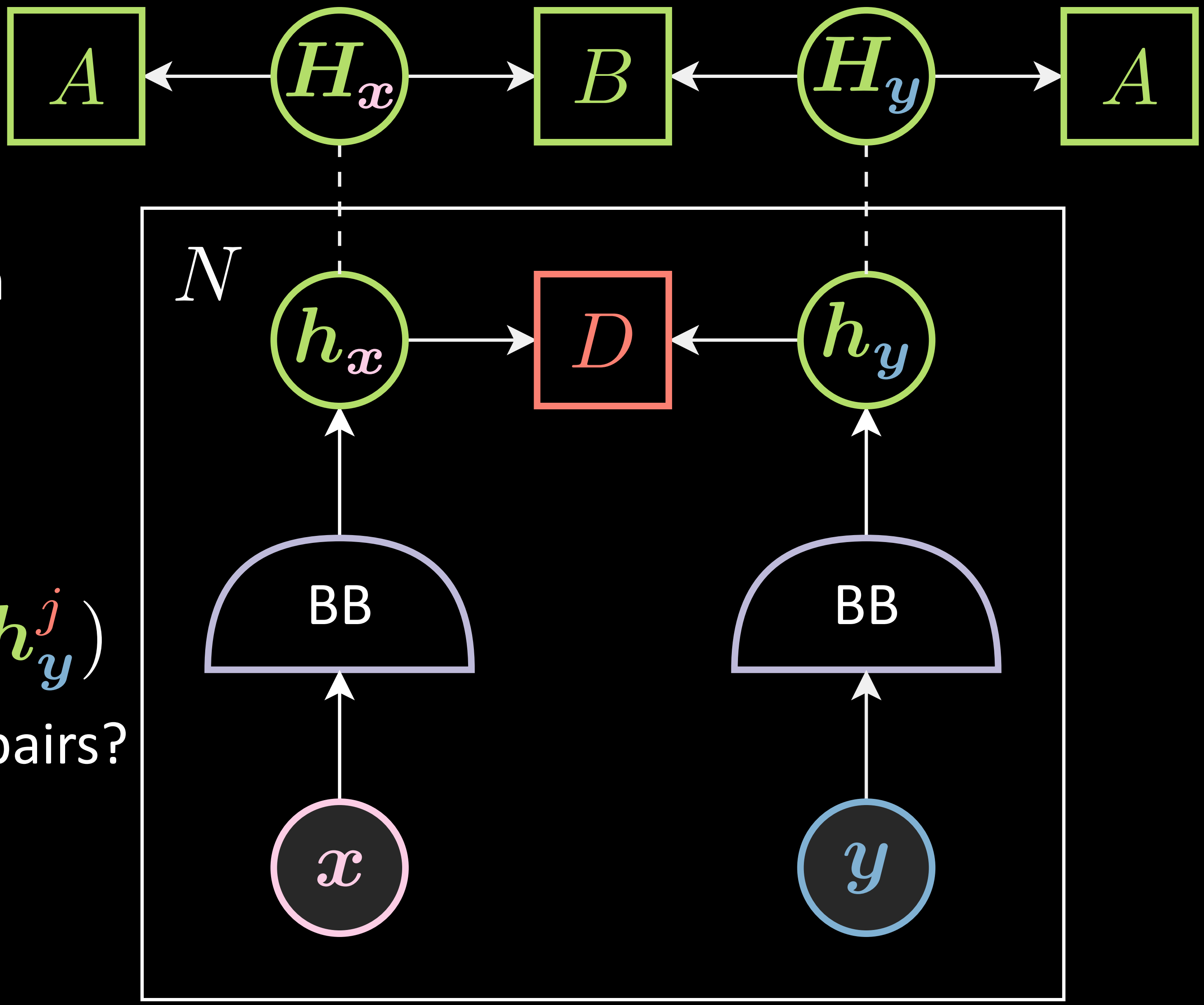
$$(h_x^i, h_y^i)$$

2. the negative pairs away

$$(h_x^i, h_x^j), (h_x^i, h_y^j), (h_y^i, h_y^j)$$

How to find a good negative pairs?

hard negative mining?

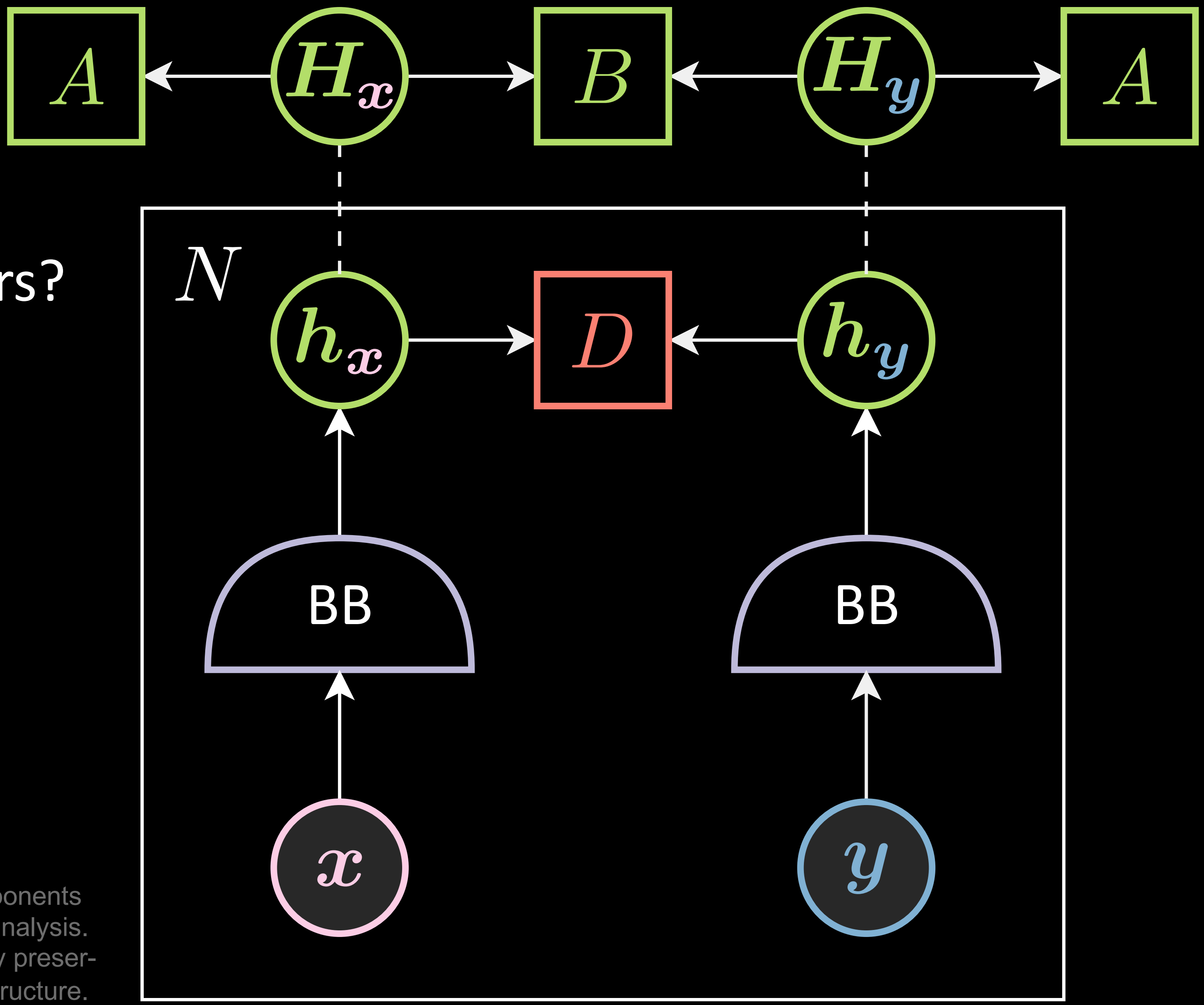


Chopra, Hadsell & LeCun (2005, June). Learning a similarity metric discriminatively, with application to face verification.
Hadsell, Chopra & LeCun (2006, June). Dimensionality reduction by learning an invariant mapping.

Contrastive Methods
SimCLR and MoCo

- How to find a good negative pairs?
- Use large batch size!

- Both SimCLR and MoCo
  use the InfoNCE loss function:



Goldberger, Hinton, Roweis & Salakhutdinov (2004). Neighbourhood components analysis.

Salakhutdinov & Hinton (2007, March). Learning a nonlinear embedding by preserving class neighbourhood structure.

Van den Oord, Li & Vinyals (2018). Representation learning with contrastive predictive coding.

Chen, Kornblith, Norouzi & Hinton (2020, November). A simple framework for contrastive learning of visual representations.

He, Fan, Wu, Xie & Girshick (2020). Momentum contrast for unsupervised visual representation learning.

The InfoNCE cost function

$$D(\boldsymbol{h}_{\boldsymbol{x}}, \boldsymbol{h}_{\boldsymbol{y}}) =$$

$$= -\log \frac{\exp(\beta \, \mathrm{sim}(\boldsymbol{h}_{\boldsymbol{x}}, \boldsymbol{h}_{\boldsymbol{y}}))}{\sum_j^N \exp\big(\beta \, \mathrm{sim}(\boldsymbol{h}_{\boldsymbol{x}}, \boldsymbol{h}_{\boldsymbol{x}}^j)\big) + \sum_j^N \exp\big(\beta \, \mathrm{sim}(\boldsymbol{h}_{\boldsymbol{x}}, \boldsymbol{h}_{\boldsymbol{y}}^j)\big)}$$

$$= -\beta \, \mathrm{sim}(\boldsymbol{h}_{\boldsymbol{x}}, \boldsymbol{h}_{\boldsymbol{y}}) + \log \left[\sum_j^N \exp\big(\beta \, \mathrm{sim}(\boldsymbol{h}_{\boldsymbol{x}}, \boldsymbol{h}_{\boldsymbol{x}}^j)\big) + \sum_j^N \exp\big(\beta \, \mathrm{sim}(\boldsymbol{h}_{\boldsymbol{x}}, \boldsymbol{h}_{\boldsymbol{y}}^j)\big) \right]$$

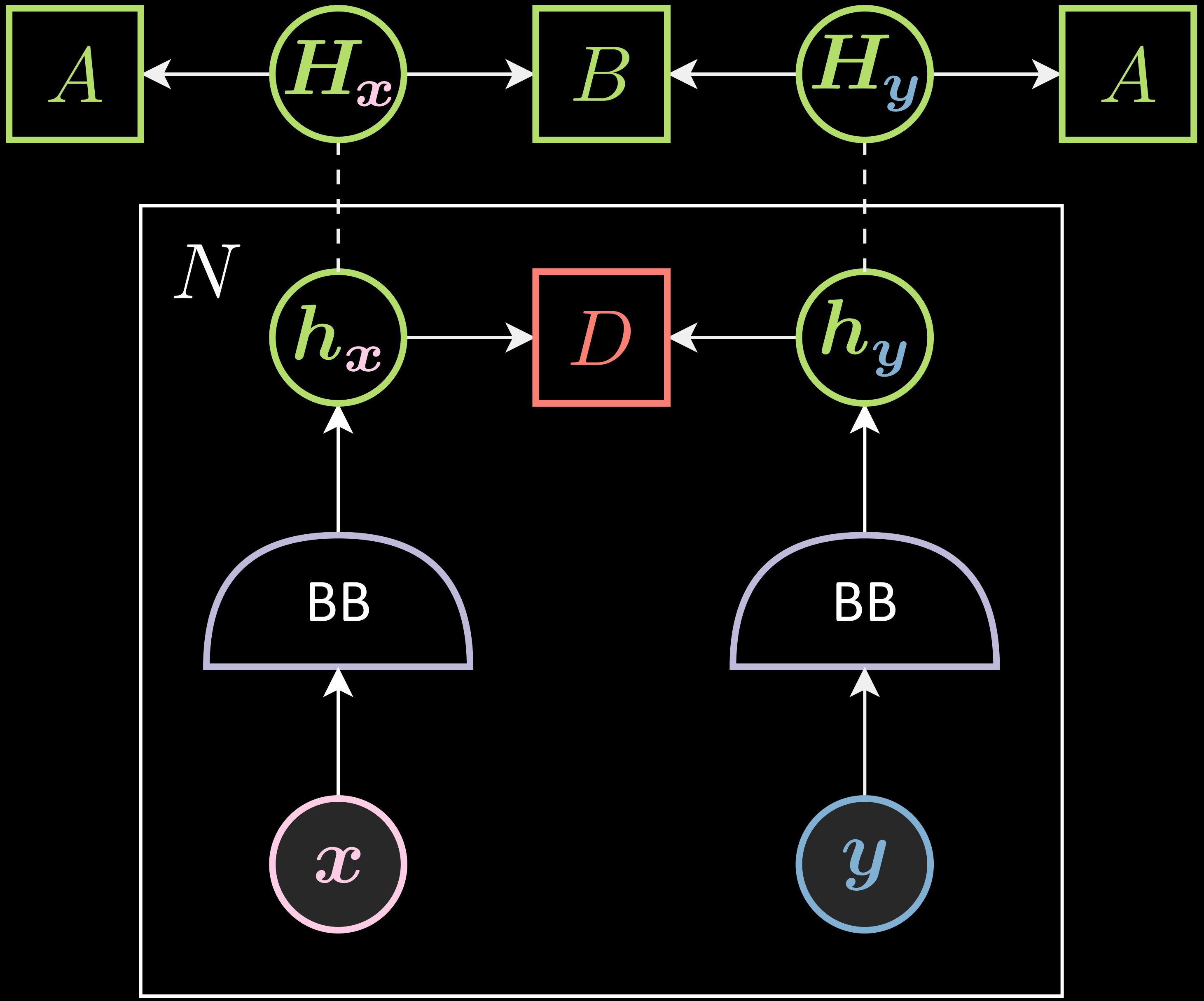
$$= -\beta \, \mathrm{sim}(\boldsymbol{h}_{\boldsymbol{x}}, \boldsymbol{h}_{\boldsymbol{y}}) + \underset{j}{\mathrm{softmax}_\beta} \left[\mathrm{sim}(\boldsymbol{h}_{\boldsymbol{x}}, \boldsymbol{h}_{\boldsymbol{x}}^j), \mathrm{sim}(\boldsymbol{h}_{\boldsymbol{x}}, \boldsymbol{h}_{\boldsymbol{y}}^j) \right]$$

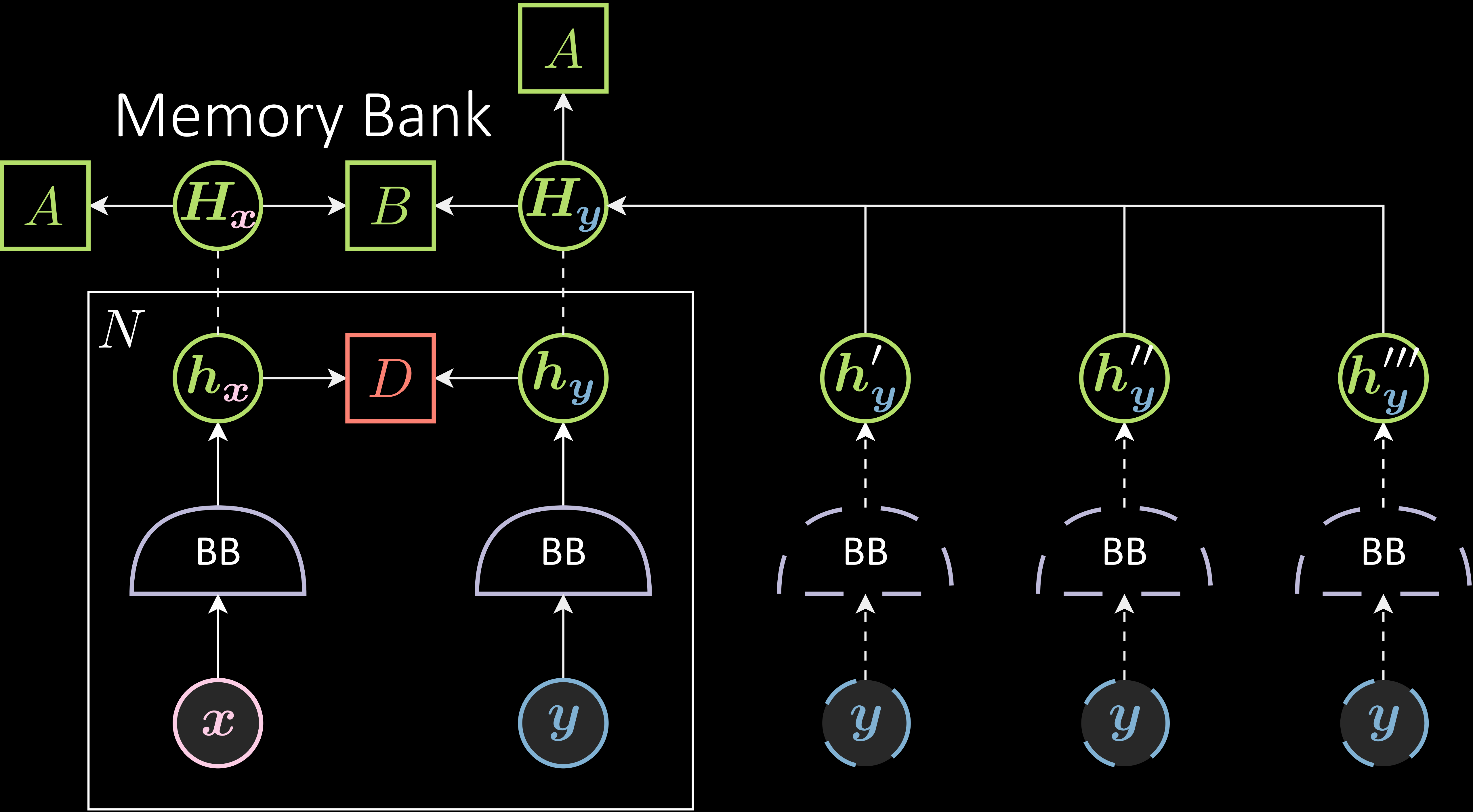
$$\mathrm{sim}(\boldsymbol{h}_{\boldsymbol{x}}, \boldsymbol{h}_{\boldsymbol{y}}) = \frac{\boldsymbol{h}_{\boldsymbol{x}}^\top \boldsymbol{h}_{\boldsymbol{y}}}{\|\boldsymbol{h}_{\boldsymbol{x}}\| \|\boldsymbol{h}_{\boldsymbol{y}}\|}$$

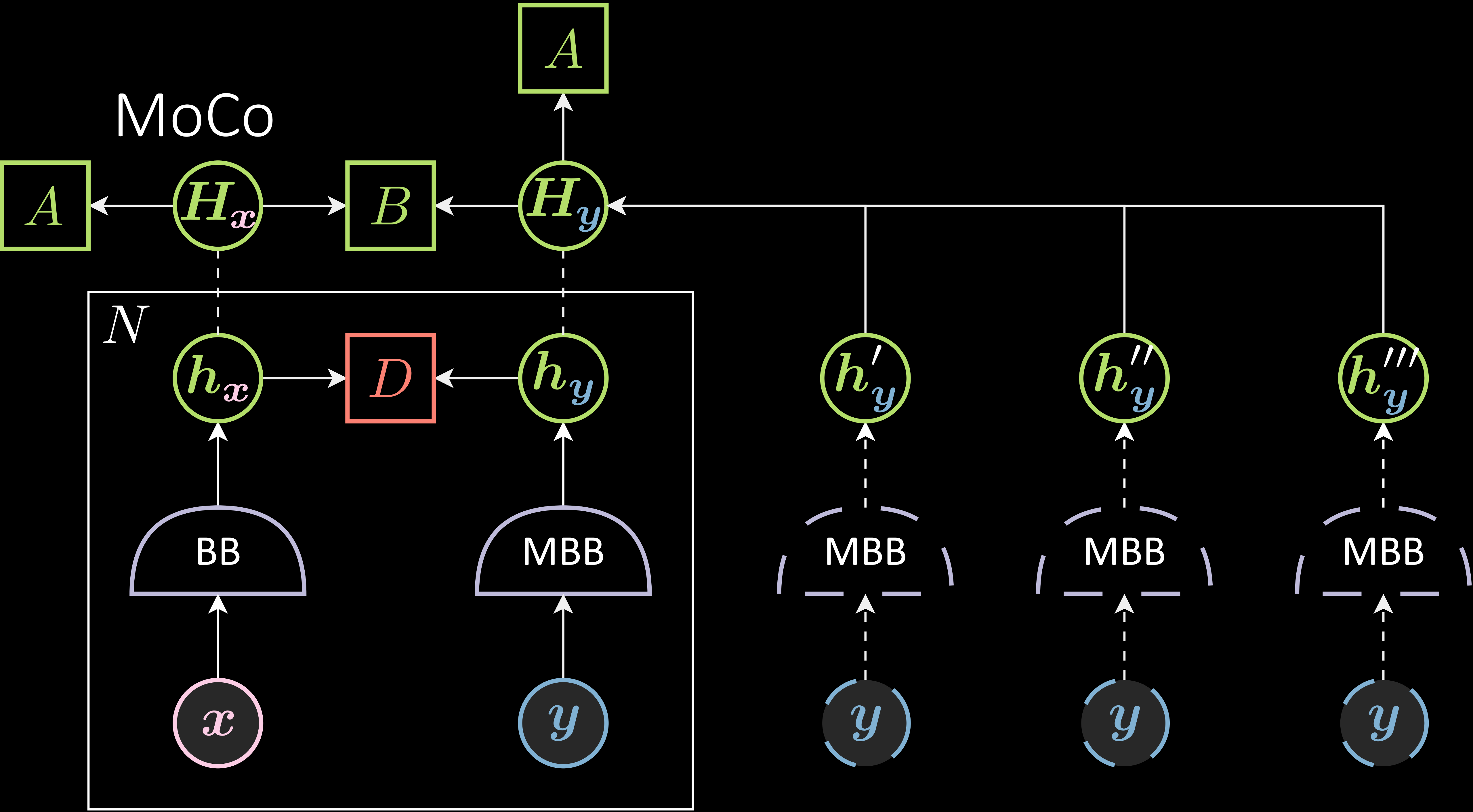
Batch size

Very large!

SimCLR N = 8192

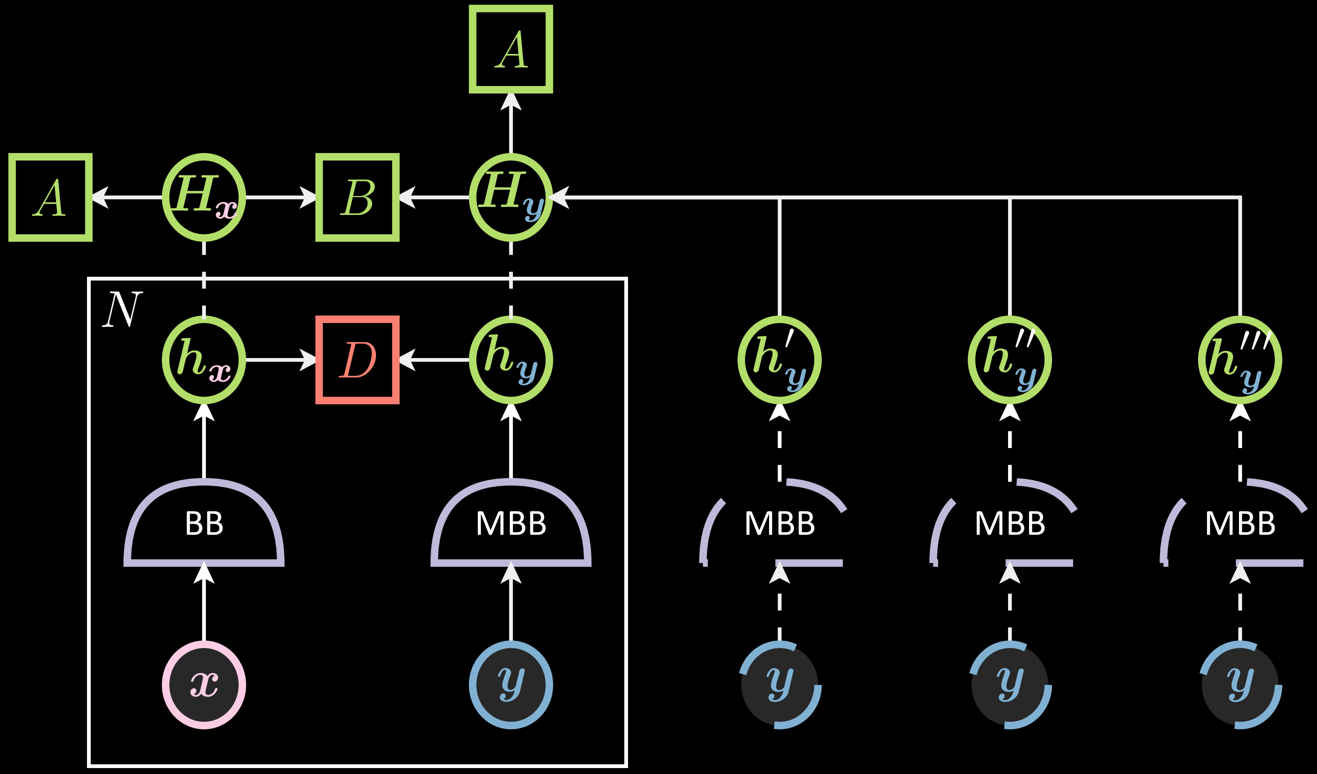



Memory Bank
A
H_x
B
H_y
A
N
h_x
D
h_y
h'_y
h''_y
h'''_y
BB
BB
BB
BB
BB
x
y
y
y
y



MoCo
A
H_x
B
H_y
A
N
h_x
D
h_y
h'_y
h''_y
h'''_y
BB
MBB
MBB
MBB
MBB
x
y
y
y
y

Contrastive Methods – MoCo



MoCo n = 256

$$\theta_{t+1} = \theta_t - \eta \nabla \theta_t$$

$$\vartheta_{t+1} = m\vartheta_t + (1-m)\theta_{t+1}$$

θ :backbone parameters
ϑ :momentum backbone parameters

Recap

What we’ve learnt so far

Quick Recap

1. Visual representation learning: pretraining + evaluation
2. Generative vs. pretext task vs. joint-embedding methods
3. Joint-embedding methods:
    1. Invariant to data augmentation
    2. Prevent trivial solution
4. Contrastive methods
    1. Hard negative mining
    2. Large negative sample pool (SimCLR vs. MoCo)

Non-contrastive methods

Prevent trivial solution without negative samples

The Disadvantages of Contrastive Methods

Contrastive methods require techniques such as:

1. weight sharing between the branches
2. batch normalization
3. feature-wise normalization
4. output quantization
5. stop gradient
6. memory banks
7. …

Non-contrastive methods and information theory

Most of non-contrastive methods

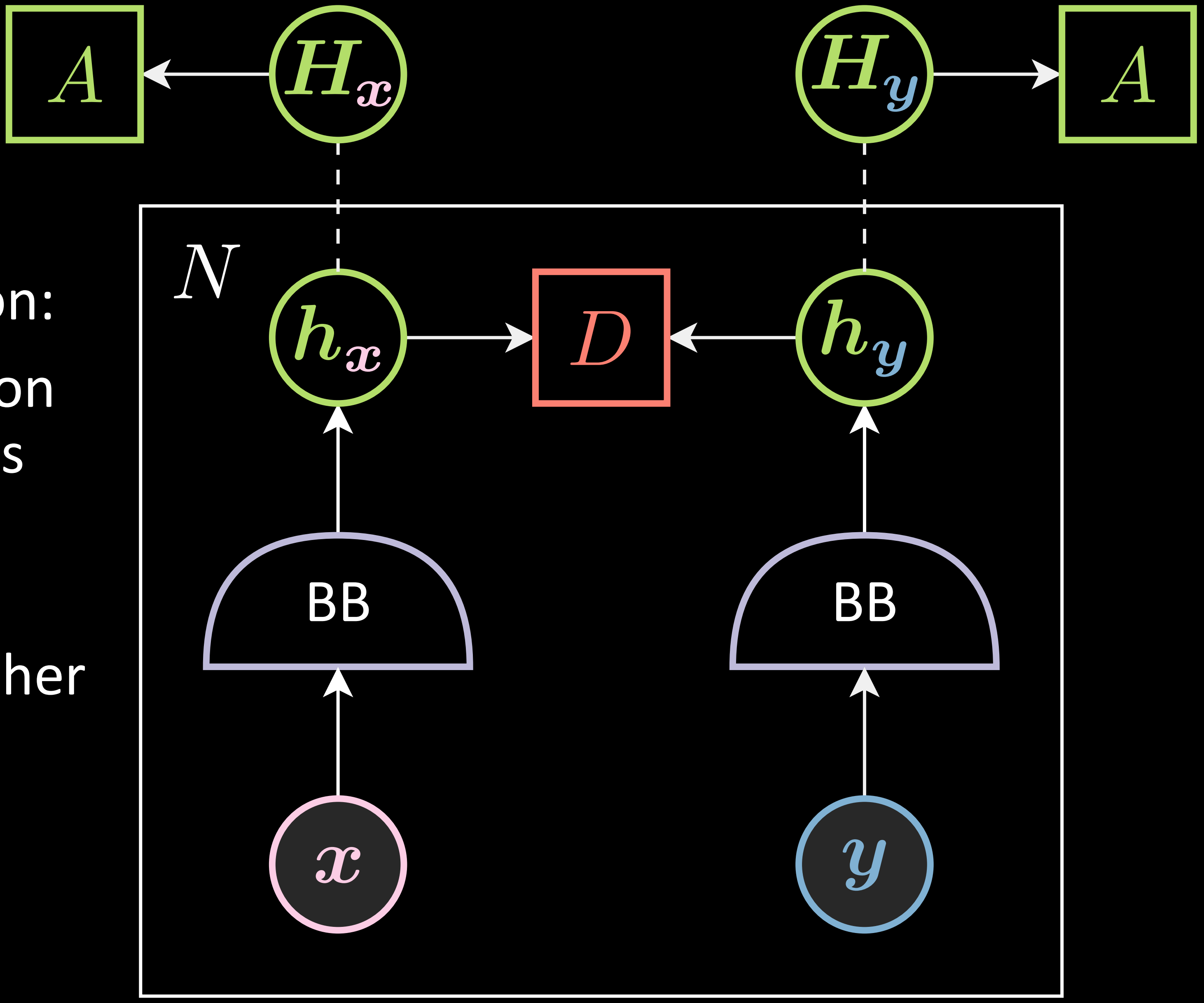
- based on information theory
  - Redundancy reduction (Barlow Twins)
  - Information maximization (VICReg)
- Don’t require special architectures

Zbontar, Jing, Misra, LeCun, & Deny. (2021). Barlow twins: Self-supervised learning via redundancy reduction.
Bardes, Ponce & LeCun (2021). Vicreg: Variance-invariance-covariance regularization for self-supervised learning.

Non-Contrastive Methods – VICReg

The Information maximization:

1. to maximize the information content of the embeddings
2. produce embedding variables that are decorrelated from each other
3. prevent an informational collapse in which the variables carry redundant information



Non-Contrastive Methods – VICReg

The loss function is pushing

1. the positive pairs closer $(\boldsymbol{h}_{\boldsymbol{x}}^{i}, \boldsymbol{h}_{\boldsymbol{y}}^{i})$
2. the variance of the embeddings large
3. The covariance of the embeddings small

$$L(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y}) = \qquad {}_{\cdot}\boldsymbol{C} = \frac{1}{N}\boldsymbol{H}_{\cdot}^{\top}\boldsymbol{H}_{\cdot}$$

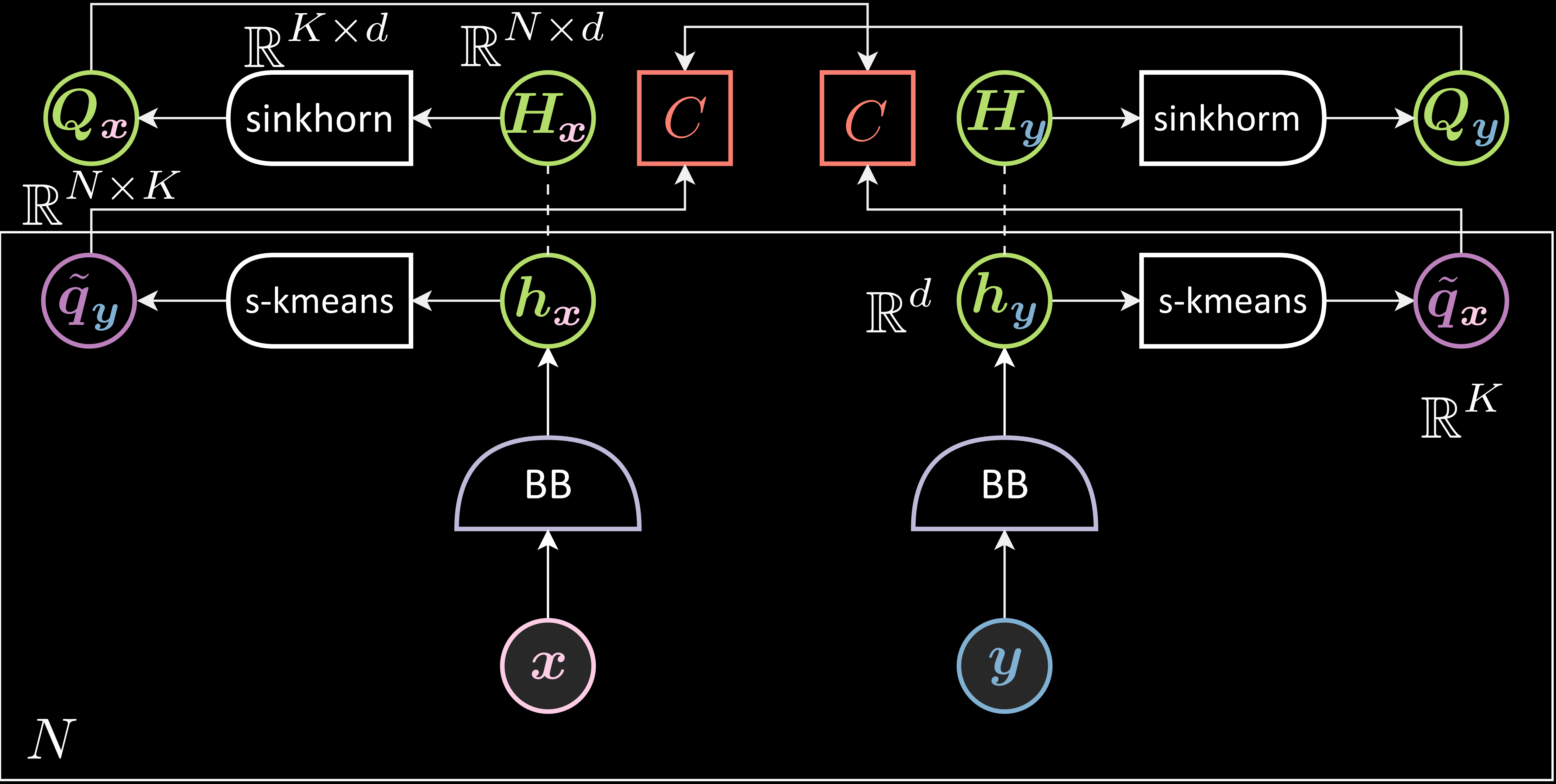
$$= \|\boldsymbol{h}_{\boldsymbol{x}} - \boldsymbol{h}_{\boldsymbol{y}}\|^2$$

$$+ \frac{1}{d}\Big[\sum_{i}^{d}(\gamma - {}_{\boldsymbol{x}}\boldsymbol{C}_{ii})^{+} + (\gamma - {}_{\boldsymbol{y}}\boldsymbol{C}_{ii})^{+}\Big]$$

$$+ \frac{1}{d}\Big[\sum_{i}^{d}\sum_{j\neq i}^{d} {}_{\boldsymbol{x}}\boldsymbol{C}_{ij}^{2} + {}_{\boldsymbol{y}}\boldsymbol{C}_{ij}^{2}\Big]$$

Clustering Methods- SwAV

Prevent trivial solution by quantizing the embedding space



Caron, Misra, Mairal, Goyal, Bojanowski, & Joulin (2020). Unsupervised learning of visual features by contrasting cluster assignments.

SwAV (II)

$$\boldsymbol{Q}_{\boldsymbol{x}} = \mathrm{sinkhorn}_{\boldsymbol{W}}(\boldsymbol{H}_{\boldsymbol{x}}) \in \mathbb{R}^{N \times K}$$

$$\boldsymbol{Q}_{\boldsymbol{x}} = [\boldsymbol{q}_{\boldsymbol{x}}^1, \dots, \boldsymbol{q}_{\boldsymbol{x}}^N]^\top$$

$$\boldsymbol{W} \in \mathbb{R}^{K \times d} : \text{dictionary}$$

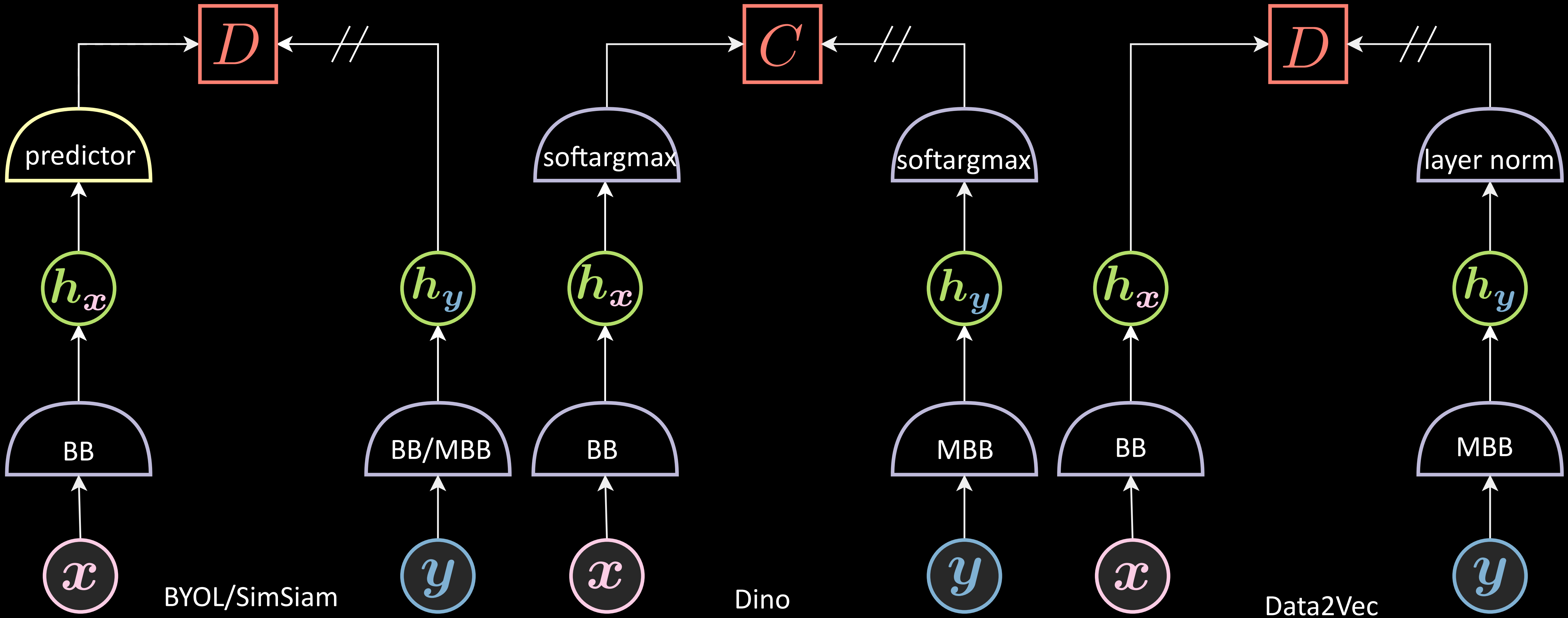
$$\tilde{\boldsymbol{q}}_{\boldsymbol{x}} = \mathrm{softargmax}_{\beta}(\boldsymbol{W}\boldsymbol{h}_{\boldsymbol{y}}) \in \mathbb{R}^{K}$$

$$F(\boldsymbol{x}, \boldsymbol{y}) = C(\boldsymbol{q}_{\boldsymbol{x}}, \tilde{\boldsymbol{q}}_{\boldsymbol{x}}) + C(\boldsymbol{q}_{\boldsymbol{y}}, \tilde{\boldsymbol{q}}_{\boldsymbol{y}})$$

Peyré, and Cuturi.(2019) Computational optimal transport: With applications to data science.

Other methods

Other Methods – BYOL, SimSiam, Dino, Data2Vec



Grill, Strub, Altché, Tallec, Richemond, Buchatskaya, ... & Valko (2020). Bootstrap your own latent-a new approach to self-supervised learning.
Chen, X., & He, K. (2021). Exploring simple siamese representation learning.
Caron, Touvron, Misra, Jégou, Mairal, Bojanowski & Joulin (2021). Emerging properties in self-supervised vision transformers.
Baevski, Hsu, Xu, Babu, Gu & Auli (2022). Data2vec: A general framework for self-supervised learning in speech, vision and language.

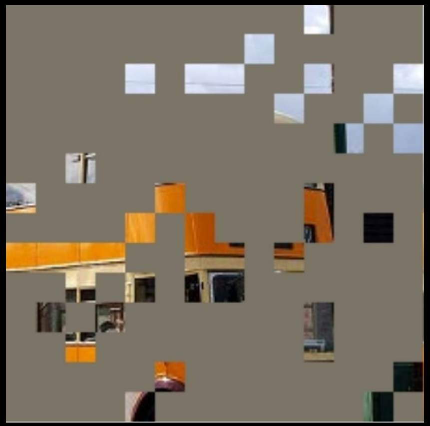
Data augmentation and network architecture

Data augmentation

The SimCLR/BYOL data augmentation:

1. Random Crop (the most critical one)
2. Flip
3. Color Jitter
4. Gaussian Blur

For traditional augmentation to masking augmentation



Network architecture

It is always better to add a two/three-layer projector/expander

Even without memory bank, momentum encoder usually helps the performance of the downstream tasks, especially with weak data augmentation