# Contrastive Multiview Coding

Chengkun Li

lichengkun0805@gmail.com

# Outline

# Contrastive Multiview Coding[1]

- Apply **Contrastive Learning** method
- Use **multiple** views from the same scene
- Aiming to learn the **representation** of data

[1] *Yonglong Tian et al.* Contrastive Multiview Coding (MIT& Google)

# Contrastive Multiview Coding[1]

- Apply **Contrastive Learning** method
- Use **multiple** views from the same scene
- Aiming to learn the **representation** of data

[1] *Yonglong Tian et al.* Contrastive Multiview Coding (MIT& Google)

# Contrastive Multiview Coding[1]

- Apply **Contrastive Learning** method
- Use **multiple** views from the same scene
- Aiming to learn the **representation** of data

[1] *Yonglong Tian et al.* Contrastive Multiview Coding (MIT& Google)

# Contrastive Multiview Coding[1]

- Apply **Contrastive Learning** method
- Use **multiple** views from the same scene
- Aiming to learn the **representation** of data

[1] *Yonglong Tian et al.* Contrastive Multiview Coding (MIT& Google)

# Outline

# Concepts & Basic Idea

**Definition of Contrastive Learning:**

A learning paradigm that learns to tell *distinctiveness*

## Definition of Contrastive Learning:

A learning paradigm that learns to (by) tell (ing) *distinctiveness*

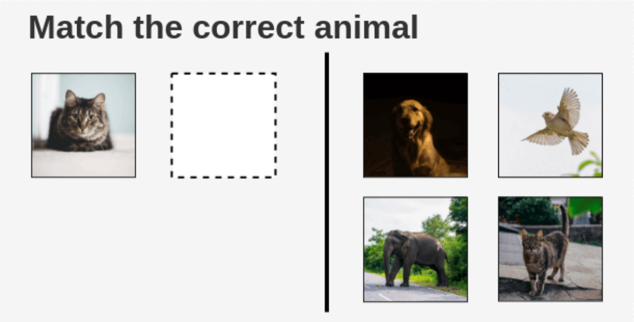
# Concepts & Basic Idea

## Definition of Contrastive Learning:

A learning paradigm that learns to (by) tell (ing) *distinctiveness*

# Intuition

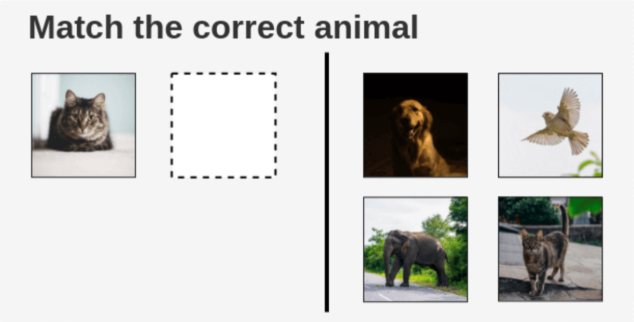
## Puzzle...




Can we teach machines this way?

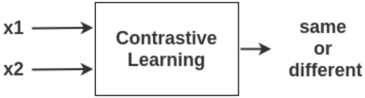
# Intuition

## Puzzle...




Can we teach machines this way?

**Goal**: Teach machines to distinguish between *similar* and *dissimilar* things



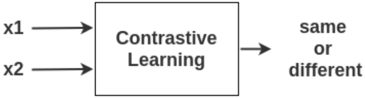

What the machines need:

1. Similar & dissimilar data
2. Ability to *represent* the image (data)
3. Ability to *quantify* if two images are similar by their representation

How?

# Problem Formulation

**Goal**: Teach machines to distinguish between *similar* and *dissimilar* things

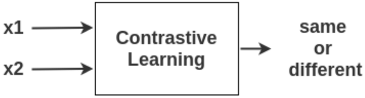



**What the machines need:**

1. Similar & dissimilar data
2. Ability to *represent* the image (data)
3. Ability to *quantify* if two images are similar by their representation

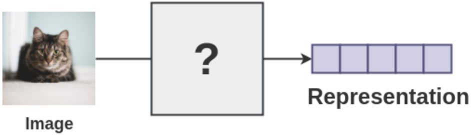
How?

**Goal**: Teach machines to distinguish between *similar* and *dissimilar* things




**What the machines need:**

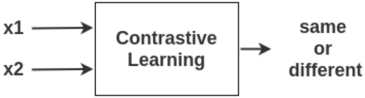
1. Similar & dissimilar data
2. Ability to *represent* the image (data)




3. Ability to *quantify* if two images are similar by their representation

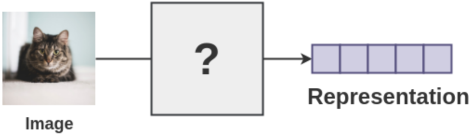
How?

**Goal**: Teach machines to distinguish between *similar* and *dissimilar* things




**What the machines need:**

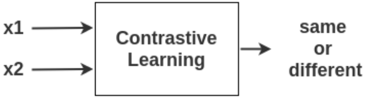
1. Similar & dissimilar data
2. Ability to *represent* the image (data)




3. Ability to *quantify* if two images are similar by their representation

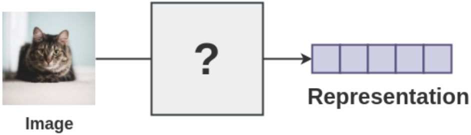
How?

**Goal**: Teach machines to distinguish between *similar* and *dissimilar* things




**What the machines need:**

1. Similar & dissimilar data
2. Ability to *represent* the image (data)



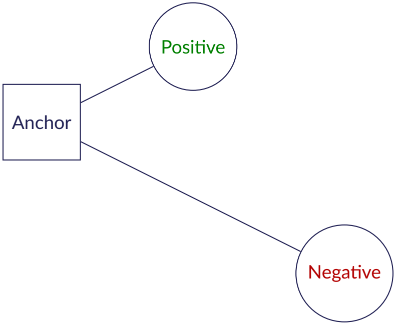

3. Ability to *quantify* if two images are similar by their representation

How?

# Similar & Dissimilar data

## Definitions

Anchor, Positive, Negative...



# Similar & Dissimilar data

## **Similar** data

1. *Supervised way*: data with same label
2. *Unsupervised way/Self-Supervised way*:
   - data augmentation
   - Multi modal
   - random cropping

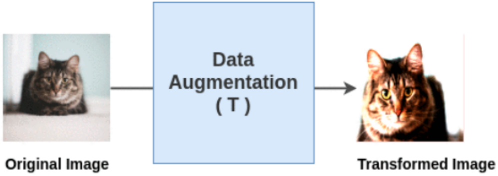
# Similar & Dissimilar data

## **Similar** data

1. *Supervised way*: data with same label
2. *Unsupervised way/Self-Supervised way*:
   - data augmentation
   - Multi modal
   - random cropping

## **Similar** data

1. *Supervised way*: data with same label
2. *Unsupervised way/Self-Supervised way*:
   - data augmentation

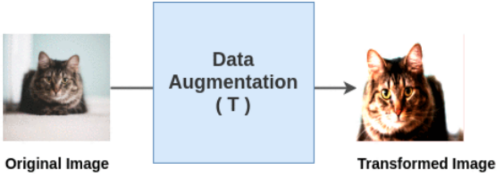



   - Multi modal
   - random cropping

# Similar & Dissimilar data

## **Similar** data

1. *Supervised way*: data with same label
2. *Unsupervised way/Self-Supervised way*:
   - data augmentation

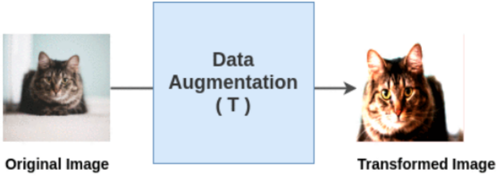



   - Multi modal
   - random cropping

# Similar & Dissimilar data

## **Similar** data

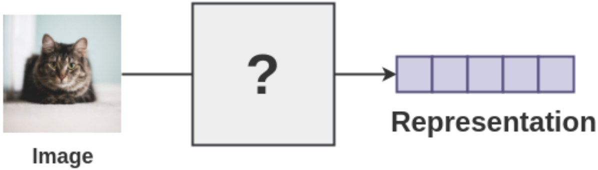
1. *Supervised way*: data with same label
2. *Unsupervised way/Self-Supervised way*:
   - data augmentation




   - Multi modal
   - random cropping

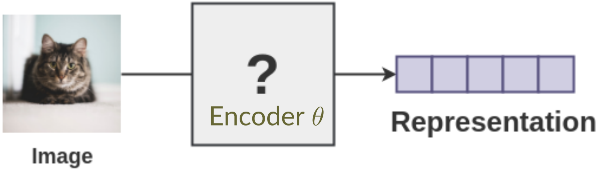
# *Represent* the data

## Data Representation




$$z_i = f_\theta(x_i)$$

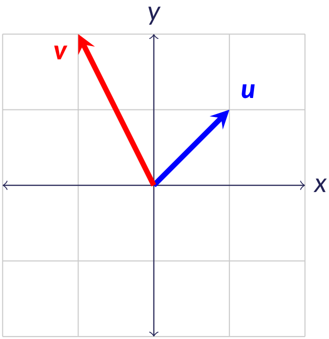
## Data Representation




$$z_i = f_\theta(x_i)$$

# Distinctiveness metric

How to quantify the distinctiveness?

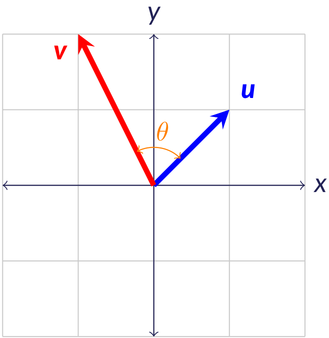



***Cosine Similarity***

$$\mathrm{sim}(u, v) = \frac{u^T v}{\|u\|\|v\|} \tag{1}$$

# Distinctiveness metric

How to quantify the distinctiveness?

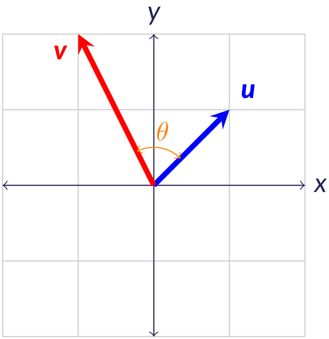



***Cosine Similarity***

$$\text{sim}(u, v) = \frac{u^T v}{\|u\| \|v\|} \tag{1}$$

# Distinctiveness metric

How to quantify the distinctiveness?




***Cosine Similarity***

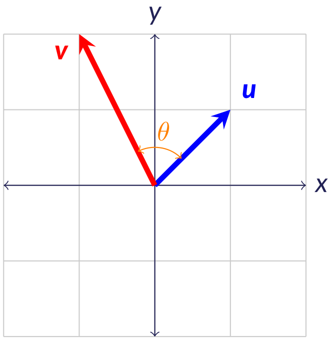
$$\mathrm{sim}(u, v) = \frac{u^T v}{\|u\| \|v\|} \quad (1)$$

sometimes, add a dynamic range controlling hyperparam $\tau$:

$$\mathrm{sim}(u, v) \cdot \frac{1}{\tau} = \frac{u^T v}{\|u\| \|v\| \tau}, \ \tau \in [-1, 1] \quad (2)$$

# Distinctiveness metric

How to quantify the distinctiveness?




***Cosine Similarity***

$$\mathrm{sim}(u, v) = \frac{u^T v}{\|u\| \|v\|} \tag{1}$$

sometimes, add a dynamic range controlling hyperparam $\tau$:

$$\mathrm{sim}(u, v) \cdot \frac{1}{\tau} = \frac{u^T v}{\|u\| \|v\| \tau} \quad , \tau \in [-1, 1] \tag{2}$$

( Some papers incorporate $\tau$ into $\mathrm{sim}(u, v)$)

# Outline

# How to train?

## Steps (self-supervised)

1. Create positive/negative (in/congruent) data pairs (e.g. data augmentations)
2. Compute Data Representation (feature extraction)
3. Compute *contrastive loss* to optimize representation in previous step

# How to train?

## Steps (self-supervised)

1. Create positive/negative (in/congruent) data pairs (e.g. data augmentations)
2. Compute Data Representation (feature extraction)
3. Compute *contrastive loss* to optimize representation in previous step

# How to train?

## Steps (self-supervised)

1. Create positive/negative (in/congruent) data pairs (e.g. data augmentations)
2. Compute Data Representation (feature extraction)
3. Compute *contrastive loss* to optimize representation in previous step

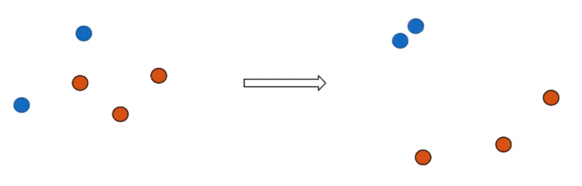
# How to train?

## Steps (self-supervised)

1. Create positive/negative (in/congruent) data pairs (e.g. data augmentations)
2. Compute Data Representation (feature extraction)
3. Compute *contrastive loss* to optimize representation in previous step

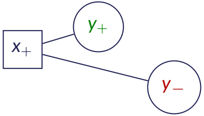
**Optimization intuition**

Pull similar pairs **together**, push dissimilar pairs **away from** each others



# How to train?

## Common Loss functions




$x_+$: Anchor; $y_+$: Positive; $y_-$: Negative

1. **Triplet margin**

$$\max\left(\|f(x_+) - f(y_+)\|^2 - \|f(x_+) - f(y_-)\|^2 + m, 0\right) \tag{3}$$

2. **NCE** loss

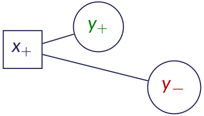
$$\log\sigma\left(\mathrm{dis}\left(x_+, y_+\right)/\tau\right) + \log\sigma\left(-\mathrm{dis}\left(x_+, y_-^i\right)/\tau\right) \tag{4}$$

3. **k-pair** loss (softmax-like)

$$-\log\frac{\exp\left(\mathrm{sim}\left(x_+, y_+\right)/\tau\right)}{\exp\left(\mathrm{sim}\left(x_+, y_+\right)/\tau\right) + \sum_{i=1}^{k}\exp\left(\mathrm{sim}\left(x_+, y_-^i\right)/\tau\right)} \tag{5}$$

# How to train?

## Common Loss functions




$x_+$: Anchor; $y_+$: Positive; $y_-$: Negative

1. **Triplet margin**

$$\max\left(\|f(x_+)-f(y_+)\|^2-\|f(x_+)-f(y_-)\|^2+m,0\right) \tag{3}$$

2. **NCE** loss

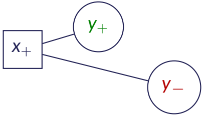
$$\log\sigma\left(\operatorname{dis}(x_+,y_+)/\tau\right)+\log\sigma\left(-\operatorname{dis}\left(x_+,y_-^i\right)/\tau\right) \tag{4}$$

3. **k-pair** loss (softmax-like)

$$-\log\frac{\exp\left(\operatorname{sim}(x_+,y_+)/\tau\right)}{\exp\left(\operatorname{sim}(x_+,y_+)/\tau\right)+\sum_{i=1}^{k}\exp\left(\operatorname{sim}\left(x_+,y_-^i\right)/\tau\right)} \tag{5}$$

# How to train?

## Common Loss functions




$x_+$: Anchor; $y_+$: Positive; $y_-$: Negative

1. **Triplet margin**

$$\max\left(\|f(x_+) - f(y_+)\|^2 - \|f(x_+) - f(y_-)\|^2 + m, 0\right) \tag{3}$$

2. **NCE** loss

$$\log\sigma\left(\text{dis}\left(x_+, y_+\right)/\tau\right) + \log\sigma\left(-\,\text{dis}\left(x_+, y_-^i\right)/\tau\right) \tag{4}$$

3. **k-pair** loss (softmax-like)

$$-\log\frac{\exp\left(\text{sim}\left(x_+, y_+\right)/\tau\right)}{\exp\left(\text{sim}\left(x_+, y_+\right)/\tau\right) + \sum_{i=1}^{k}\exp\left(\text{sim}\left(x_+, y_-^i\right)/\tau\right)} \tag{5}$$

# Outline

# Outline

# Contrastive Multiview Coding[2]

- Apply Contrastive Learning method
- Use **multiple** views from the same scene
- Aiming to learn the **representation** of data

[2] *Yonglong Tian et al.* Contrastive Multiview Coding

# Contrastive Multiview Coding[2]

- Apply Contrastive Learning method
- Use **multiple** views from the same scene
- Aiming to learn the **representation** of data

[2] *Yonglong Tian et al.* Contrastive Multiview Coding

# Contrastive Multiview Coding[2]

- Apply Contrastive Learning method
- Use **multiple** views from the same scene
- Aiming to learn the **representation** of data

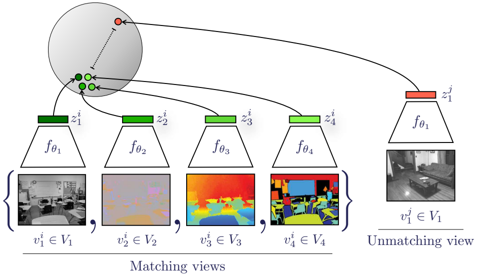
[2] *Yonglong Tian et al.* Contrastive Multiview Coding

# Contrastive Multiview Coding[2]

- Apply Contrastive Learning method
- Use **multiple** views from the same scene
- Aiming to learn the **representation** of data

[2] *Yonglong Tian et al.* Contrastive Multiview Coding

Comparing with the previous self-supervised example,

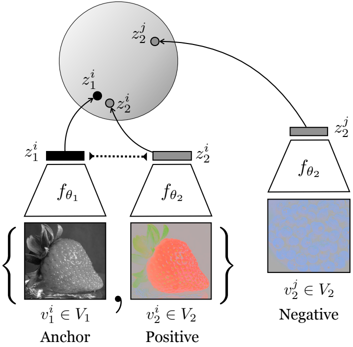



CMC...

1. data augmentation $\rightarrow$ views
2. two sample data $\rightarrow$ multiple views

# Outline

## Notations




**Recall: sim function**

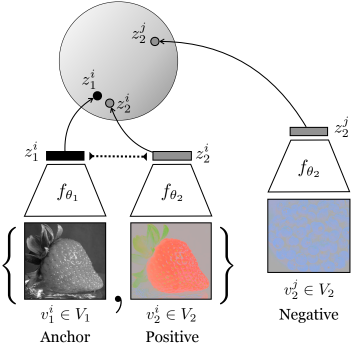
$$\text{sim}(u, v) = \frac{u^T v}{\|u\|\|v\|}$$

- **Views**: $V_1, V_2, \cdots, V_N$
- **Data in a view**: $v_a^b$
- **Positives**: $\{v_1^i, v_2^i\}_{i=1}^N$
- **Negatives**: $\{v_1^i, v_2^j\}$, $i, j \in N$
- **Discriminating function**: $h_\theta(\cdot)$

$$h_\theta(\{v_1, v_2\}) = \exp\left(\frac{f_{\theta_1}(v_1) \cdot f_{\theta_2}(v_2)}{\|f_{\theta_1}(v_1)\| \cdot \|f_{\theta_2}(v_2)\|} \cdot \frac{1}{\tau}\right) \quad (6)$$

- **Encoders**: $\theta_1, \theta_2, \cdots, \theta_N$

## Notations




**Recall: sim function**

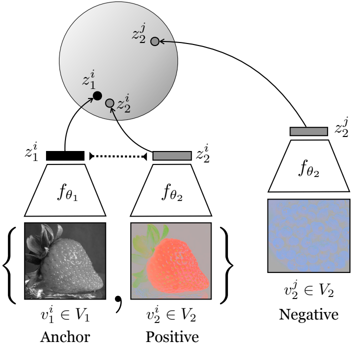
$$\text{sim}(u, v) = \frac{u^T v}{\|u\|\|v\|}$$

- **Views**: $V_1, V_2, \cdots, V_N$
- **Data in a view**: $v_a^b$
- **Positives**: $\{v_1^i, v_2^i\}_{i=1}^N$
- **Negatives**: $\{v_1^i, v_2^j\}, \ i, j \in N$
- **Discriminating function**: $h_\theta(\cdot)$

$$h_\theta(\{v_1, v_2\}) = \exp\left(\frac{f_{\theta_1}(v_1) \cdot f_{\theta_2}(v_2)}{\|f_{\theta_1}(v_1)\| \cdot \|f_{\theta_2}(v_2)\|} \cdot \frac{1}{\tau}\right) \quad (6)$$

- **Encoders**: $\theta_1, \theta_2, \cdots, \theta_N$

## Notations




**Recall: sim function**

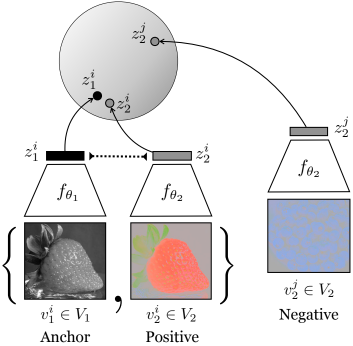
$$\text{sim}(u, v) = \frac{u^T v}{\|u\| \|v\|}$$

- **Views:** $V_1, V_2, \cdots, V_N$
- **Data in a view:** $v_a^b$
- **Positives:** $\{v_1^i, v_2^i\}_{i=1}^N$
- **Negatives:** $\{v_1^i, v_2^j\}, \; i, j \in N$
- **Discriminating function:** $h_\theta(\cdot)$

$$h_\theta(\{v_1, v_2\}) = \exp\left(\frac{f_{\theta_1}(v_1) \cdot f_{\theta_2}(v_2)}{\|f_{\theta_1}(v_1)\| \cdot \|f_{\theta_2}(v_2)\|} \cdot \frac{1}{\tau}\right) \quad (6)$$

- **Encoders:** $\theta_1, \theta_2, \cdots, \theta_N$

## Notations




**Recall: sim function**

$$\mathrm{sim}(u, v) = \frac{u^T v}{\|u\| \|v\|}$$

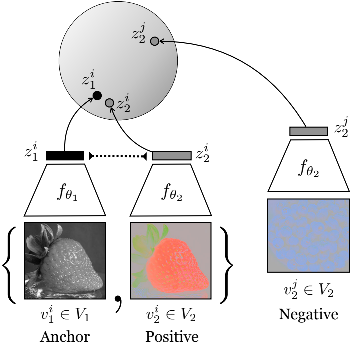
- **Views**: $V_1, V_2, \cdots, V_N$
- **Data in a view**: $v_a^b$
- **Positives**: $\{v_1^i, v_2^i\}_{i=1}^N$
- **Negatives**: $\{v_1^i, v_2^j\},\ i, j \in N$
- **Discriminating function**: $h_\theta(\cdot)$

$$h_\theta(\{v_1, v_2\}) = \exp\left(\frac{f_{\theta_1}(v_1) \cdot f_{\theta_2}(v_2)}{\|f_{\theta_1}(v_1)\| \cdot \|f_{\theta_2}(v_2)\|} \cdot \frac{1}{\tau}\right) \quad (6)$$

- **Encoders**: $\theta_1, \theta_2, \cdots, \theta_N$

# Concepts & Basic Idea

## Notations




**Recall: sim function**

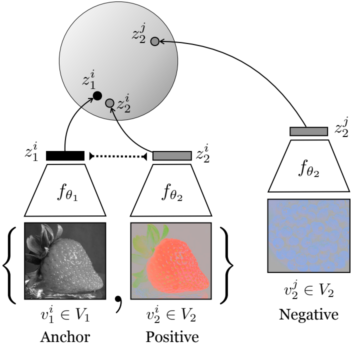
$$\mathrm{sim}(u, v) = \frac{u^T v}{\|u\| \|v\|}$$

- **Views**: $V_1, V_2, \cdots, V_N$
- **Data in a view**: $v_a^b$
- **Positives**: $\{v_1^i, v_2^i\}_{i=1}^N$
- **Negatives**: $\{v_1^i, v_2^j\},\ i, j \in N$
- **Discriminating function**: $h_\theta(\cdot)$

$$h_\theta(\{v_1, v_2\}) = \exp\left(\frac{f_{\theta_1}(v_1) \cdot f_{\theta_2}(v_2)}{\|f_{\theta_1}(v_1)\| \cdot \|f_{\theta_2}(v_2)\|} \cdot \frac{1}{\tau}\right) \quad (6)$$

- **Encoders**: $\theta_1, \theta_2, \cdots, \theta_N$

## Notations




**Recall: sim function**

$$\mathrm{sim}(u, v) = \frac{u^T v}{\|u\|\|v\|}$$

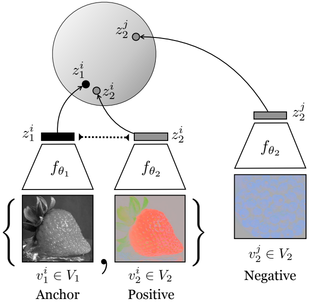
- **Views**: $V_1, V_2, \cdots, V_N$
- **Data in a view**: $v_a^b$
- **Positives**: $\{v_1^i, v_2^i\}_{i=1}^N$
- **Negatives**: $\{v_1^i, v_2^j\},\ i, j \in N$
- **Discriminating function**: $h_\theta(\cdot)$

$$h_\theta(\{v_1, v_2\}) = \exp\left(\frac{f_{\theta_1}(v_1) \cdot f_{\theta_2}(v_2)}{\|f_{\theta_1}(v_1)\| \cdot \|f_{\theta_2}(v_2)\|} \cdot \frac{1}{\tau}\right) \quad (6)$$

- **Encoders:** $\theta_1, \theta_2, \cdots, \theta_N$

# Outline

# CMC with two views




## k-pair loss

$$\ell = -\log \frac{\exp\left(\mathrm{sim}\left(x_+, y_+\right)/\tau\right)}{\exp\left(\mathrm{sim}\left(x_+, y_+\right)/\tau\right) + \sum_{i=1}^{k} \exp\left(\mathrm{sim}\left(x_+, y_-^i\right)/\tau\right)}$$

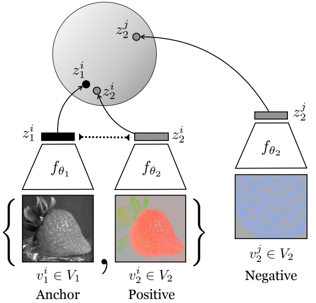
Contrastive loss between $V_1$ (anchor view), $V_2$

$$\mathcal{L}_{contrast}^{V_1, V_2} = - \mathop{\mathbb{E}}_{\{v_1^1, v_2^1, \ldots, v_2^{k+1}\}} \left[ \log \frac{h_\theta(\{v_1^1, v_2^1\})}{\sum_{j=1}^{k+1} h_\theta(\{v_1^1, v_2^j\})} \right] \quad (7)$$

Total loss for two views:

$$\mathcal{L}\left(V_1, V_2\right) = \mathcal{L}_{contrast}^{V_1, V_2} + \mathcal{L}_{contrast}^{V_2, V_1} \quad (8)$$

# CMC with two views




**k-pair loss**

$$\ell = -\log \frac{\exp\left(\operatorname{sim}\left(x_+, y_+\right)/\tau\right)}{\exp\left(\operatorname{sim}\left(x_+, y_+\right)/\tau\right) + \sum_{i=1}^{k} \exp\left(\operatorname{sim}\left(x_+, y_-^i\right)/\tau\right)}$$

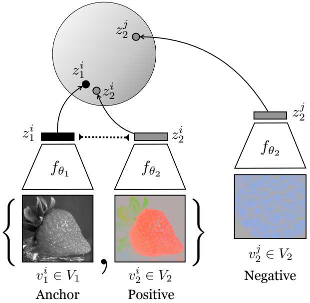
Contrastive loss between $V_1$ (anchor view), $V_2$

$$\mathcal{L}_{contrast}^{V_1,V_2} = -\mathop{\mathbb{E}}_{\{v_1^1, v_2^1, \ldots, v_2^{k+1}\}} \left[\log \frac{h_\theta(\{v_1^1, v_2^1\})}{\sum_{j=1}^{k+1} h_\theta(\{v_1^1, v_2^j\})}\right] \tag{7}$$

Total loss for two views:

$$\mathcal{L}(V_1, V_2) = \mathcal{L}_{contrast}^{V_1,V_2} + \mathcal{L}_{contrast}^{V_2,V_1} \tag{8}$$

# CMC with two views




## k-pair loss

$$\ell = -\log \frac{\exp\left(\mathrm{sim}\left(x_+, y_+\right)/\tau\right)}{\exp\left(\mathrm{sim}\left(x_+, y_+\right)/\tau\right) + \sum_{i=1}^{k} \exp\left(\mathrm{sim}\left(x_+, y_-^i\right)/\tau\right)}$$

Contrastive loss between $V_1$ (anchor view), $V_2$

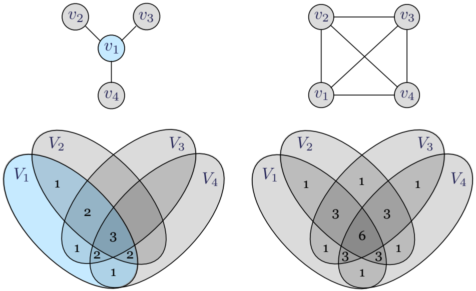
$$\mathcal{L}_{contrast}^{V_1,V_2} = -\mathop{\mathbb{E}}_{\{v_1^1, v_2^1, \ldots, v_2^{k+1}\}} \left[\log \frac{h_\theta(\{v_1^1, v_2^1\})}{\sum_{j=1}^{k+1} h_\theta(\{v_1^1, v_2^j\})}\right] \tag{7}$$

Total loss for two views:

$$\mathcal{L}\left(V_1, V_2\right) = \mathcal{L}_{\text{contrast}}^{V_1,V_2} + \mathcal{L}_{\text{contrast}}^{V_2,V_1} \tag{8}$$

# Outline

# CMC with Multiple Views

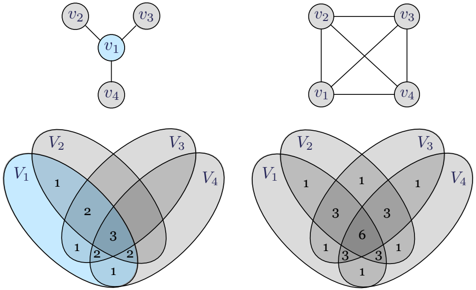



(a) Core View (b) Full Graph

$$\mathcal{L}_C = \sum_{j=2}^{M} \mathcal{L}(V_1, V_j) \tag{9}$$

$$\mathcal{L}_F = \sum_{1 \leq i < j \leq M} \mathcal{L}(V_i, V_j) \tag{10}$$

# CMC with Multiple Views

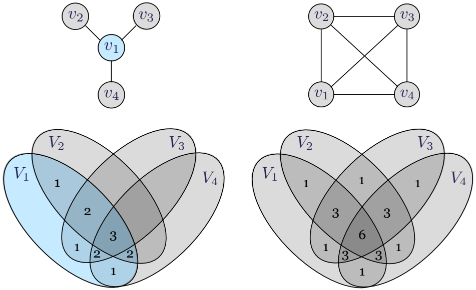



(a) Core View (b) Full Graph

$$\mathcal{L}_C = \sum_{j=2}^{M} \mathcal{L}(V_1, V_j) \tag{9}$$

$$\mathcal{L}_F = \sum_{1 \le i < j \le M} \mathcal{L}(V_i, V_j) \tag{10}$$

# CMC with Multiple Views




(a) Core View (b) Full Graph

$$\mathcal{L}_C = \sum_{j=2}^{M} \mathcal{L}(V_1, V_j) \tag{9}$$

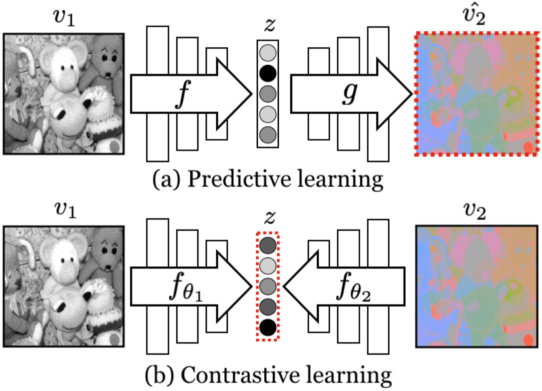
$$\mathcal{L}_F = \sum_{1 \le i < j \le M} \mathcal{L}(V_i, V_j) \tag{10}$$

# Outline

# Why CMC?

- CMC vs Encoder-Decoder
- CMC vs Supervised classifier

# CMC vs Encoder-Decoder




For **predictive** learning:

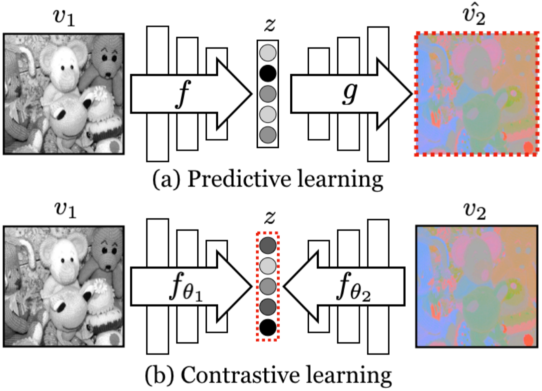
1. $z = f(v_1)$ and $\hat{v}_2 = g(z)$
2. Train $f, g$ to make $\hat{v}_2$ **closer to** $v_2$ (at output space)

For **contrastive** learning:

1. $z_1 = f_{\theta_1}(v_1), z_2 = f_{\theta_2}(v_2)$
2. Train $f_{\theta_1}, f_{\theta_2}$ to make $z_1$ **closer to** $z_2$ (at latent space)

Pixel wise $\mathcal{L}_1, \mathcal{L}_2$ loss model complex structure poorly

# CMC vs Encoder-Decoder




For **predictive** learning:

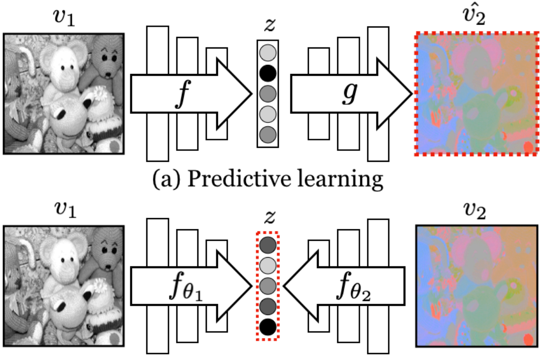
1. $z = f(v_1)$ and $\hat{v}_2 = g(z)$
2. Train $f, g$ to make $\hat{v}_2$ **closer to** $v_2$ (at output space)

For **contrastive** learning:

1. $z_1 = f_{\theta_1}(v_1), z_2 = f_{\theta_2}(v_2)$
2. Train $f_{\theta_1}, f_{\theta_2}$ to make $z_1$ **closer to** $z_2$ (at latent space)

Pixel wise $\mathcal{L}_1, \mathcal{L}_2$ loss model complex structure poorly

# CMC vs Encoder-Decoder



(a) Predictive learning
(b) Contrastive learning

For **predictive** learning:

1. $z = f(v_1)$ and $\hat{v}_2 = g(z)$
2. Train $f, g$ to make $\hat{v}_2$ **closer to** $v_2$ (at output space)

For **contrastive** learning:

1. $z_1 = f_{\theta_1}(v_1)$, $z_2 = f_{\theta_2}(v_2)$
2. Train $f_{\theta_1}, f_{\theta_2}$ to make $z_1$ **closer to** $z_2$ (at latent space)

Pixel wise $\mathcal{L}_1, \mathcal{L}_2$ loss model complex structure poorly

# CMC vs Supervised classifier

## Example

Senses

- **Vision**: dog with a mask?
- **Acoustic**: ”bark, bark”
- **Texture**: furry

Mutual Information

# CMC vs Supervised classifier

## Example

Senses

- **Vision**: dog with a mask?
- **Acoustic**: ”bark, bark”
- **Texture**: furry

Mutual Information

# CMC vs Supervised classifier

## Example

Senses

- **Vision**: dog with a mask?
- **Acoustic**: ”bark, bark”
- **Texture**: furry

Mutual Information

# CMC vs Supervised classifier

## Example

Senses

- **Vision**: dog with a mask?
- **Acoustic**: ”bark, bark”
- **Texture**: furry

Mutual Information

# CMC vs Supervised classifier

## Example

Senses

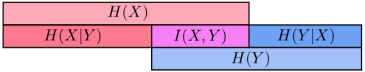
- **Vision**: dog with a mask?
- **Acoustic**: ”bark, bark”
- **Texture**: furry

Mutual Information

# Outline

# Relationship with Mutual Information

## Mutual Information[3]

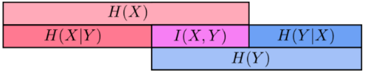



$$I(X;Y) = H(X) - H(X|Y) \tag{11}$$

$$\mathrm{I}(X;Y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} p_{(X,Y)}(x,y) \log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)}\right) dxdy \tag{12}$$

[3] Visual Information Theory *Christopher Olah*

## Mutual Information[3]



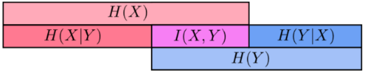

$$I(X;Y) = H(X) - H(X|Y) \tag{11}$$

$$\mathrm{I}(X;Y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} p_{(X,Y)}(x,y) \log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)}\right) dxdy \tag{12}$$

[3] Visual Information Theory *Christopher Olah*

# Relationship with Mutual Information

## Mutual Information[3]




$$I(X;Y) = H(X) - H(X|Y) \tag{11}$$

$$\mathrm{I}(X;Y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} p_{(X,Y)}(x,y) \log \left( \frac{p_{(X,Y)}(x,y)}{p_X(x) p_Y(y)} \right) dxdy \tag{12}$$

[3] Visual Information Theory *Christopher Olah*

# Relationship with Mutual Information

### *Mutual Information*

$$\mathrm{I}(X;Y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} p_{(X,Y)}(x,y) \log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)}\right) dxdy$$

Relationship between $\mathcal{L}_{contrast}$ and MI, $k$ is number of negatives

$$\begin{aligned}\mathcal{L}_{contrast} &\geq \log(k) - \mathop{\mathbb{E}}_{(z_1,z_2)\sim p_{z_1,z_2}(\cdot)} \log\left[\frac{p(z_1,z_2)}{p(z_1)p(z_2)}\right] \\ &= \log(k) - I(z_1;z_2)\end{aligned}$$

we get[4]

$$I(v_i;v_j) \geq I(z_i;z_j) \geq \log(k) - \mathcal{L}_{\text{contrast}}$$

[4] Similar idea also appears in *Oord et al.* Representation learning with contrastive predictive coding

# Relationship with Mutual Information

### *Mutual Information*

$$\mathrm{I}(X;Y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} p_{(X,Y)}(x,y) \log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)}\right) dxdy$$

Relationship between $\mathcal{L}_{contrast}$ and MI, $k$ is number of negatives

$$\begin{aligned} \mathcal{L}_{contrast} &\geq \log(k) - \mathop{\mathbb{E}}_{(z_1,z_2)\sim p_{z_1,z_2}(\cdot)} \log\left[\frac{p(z_1,z_2)}{p(z_1)p(z_2)}\right] \\ &= \log(k) - I(z_1;z_2) \end{aligned}$$

we get[4]

$$I(v_i;v_j) \geq I(z_i;z_j) \geq \log(k) - \mathcal{L}_{\text{contrast}}$$

[4] Similar idea also appears in *Oord et al.* Representation learning with contrastive predictive coding

# Relationship with Mutual Information

### Mutual Information

$$\mathrm{I}(X;Y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} p_{(X,Y)}(x,y) \log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)}\right) dxdy$$

Relationship between $\mathcal{L}_{contrast}$ and MI, $k$ is number of negatives

$$\begin{aligned}\mathcal{L}_{contrast} &\geq \log(k) - \mathop{\mathbb{E}}_{(z_1,z_2)\sim p_{z_1,z_2}(\cdot)} \log\left[\frac{p(z_1,z_2)}{p(z_1)p(z_2)}\right] \\ &= \log(k) - I(z_1;z_2)\end{aligned}$$

we get[4]

$$I(v_i;v_j) \geq I(z_i;z_j) \geq \log(k) - \mathcal{L}_{contrast}$$

[4] Similar idea also appears in *Oord et al.* Representation learning with contrastive predictive coding

# Relationship with Mutual Information

Better performance with higher MI?

$$I\left(z_i; z_j\right) \geq \log(k) - \mathcal{L}_{\text{contrast}}$$

# Relationship with Mutual Information

Better performance with higher MI?

$$I(z_i; z_j) \geq \log(k) - \mathcal{L}_{\text{contrast}}$$

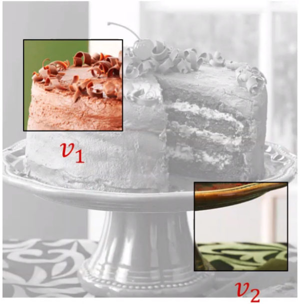
# Relationship with Mutual Information

Better performance with higher MI?

$$I(z_i; z_j) \geq \log(k) - \mathcal{L}_{\text{contrast}}$$

**Not exactly**

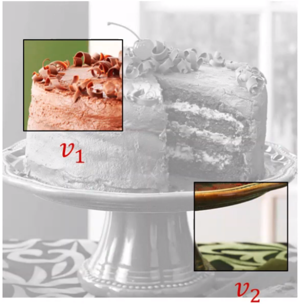
# Reducing $I(v_1; v_2)$ with Spatial Distance




**Experiment setup:**

- Two patches start at $(x, y)$ and $(x + d, y + d)$
- Train a linear classifier on pre-trained CMC representations
- Test classification accuracy

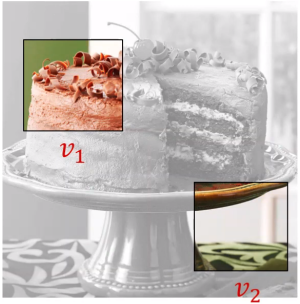
# Reducing $I(v_1; v_2)$ with Spatial Distance




**Experiment setup**:

- Two patches start at $(x, y)$ and $(x + d, y + d)$
- Train a linear classifier on pre-trained CMC representations
- Test classification accuracy

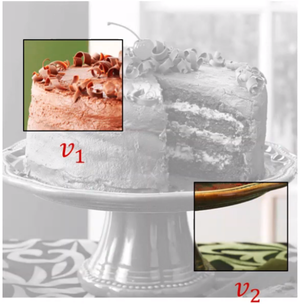
# Reducing $I(v_1; v_2)$ with Spatial Distance




**Experiment setup**:

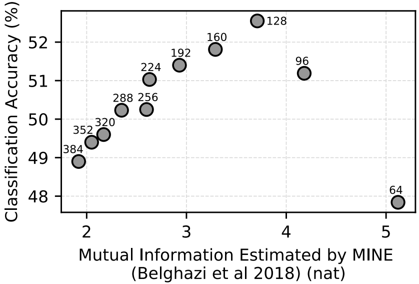
- Two patches start at $(x, y)$ and $(x + d, y + d)$
- Train a linear classifier on pre-trained CMC representations
- Test classification accuracy

# Reducing $I(v_1; v_2)$ with Spatial Distance




**Experiment setup:**

- Two patches start at $(x, y)$ and $(x + d, y + d)$
- Train a linear classifier on pre-trained CMC representations
- Test classification accuracy



# Relationship with Mutual Information

## Why?[5]

downstream task $y$, data $x$

Too much information introduces task-irrelevant noise (nuisance)

[1] *Yonglong Tian et al.* What Makes for Good Views for Contrastive Learning?

# Relationship with Mutual Information

## Why?[5]

downstream task y, data x

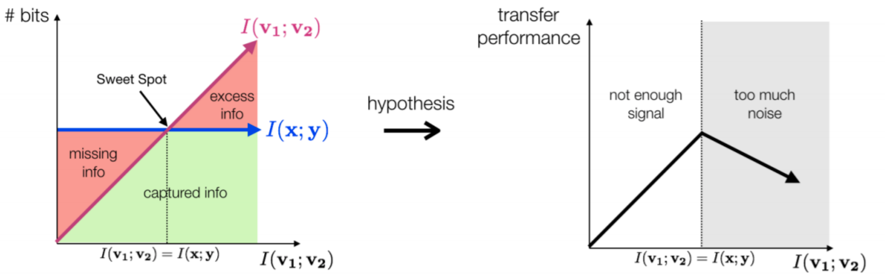
Too much information introduces task-irrelevant noise (nuisance)

[1] *Yonglong Tian et al.* What Makes for Good Views for Contrastive Learning?

# Relationship with Mutual Information

## Why?[5]

downstream task y, data x



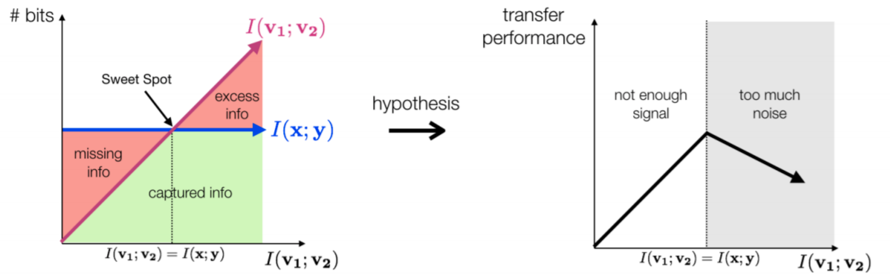

Too much information introduces task-irrelevant noise (nuisance)

[1] *Yonglong Tian et al.* What Makes for Good Views for Contrastive Learning?

# Relationship with Mutual Information

## Why?[5]

downstream task y, data x




Too much information introduces task-irrelevant noise (nuisance)

[1] *Yonglong Tian et al.* What Makes for Good Views for Contrastive Learning?

# Outline

# Conclusions & Inspirations

- CMC enables the learning of unsupervised representations from **multiple views** of datasets
- Mutual Information between views/modalities should be considered
- Contrastive Methods could be leveraged with/instead of Predictive Methods[6]
- ...

[6] Contrastive Learning for Unpaired Image-to-Image Translation (ECCV 2020)

# Conclusions & Inspirations

- CMC enables the learning of unsupervised representations from **multiple views** of datasets
- Mutual Information between views/modalities should be considered
- Contrastive Methods could be leveraged with/instead of Predictive Methods[6]
- ...

[6] Contrastive Learning for Unpaired Image-to-Image Translation (ECCV 2020)

# Conclusions & Inspirations

- CMC enables the learning of unsupervised representations from **multiple views** of datasets
- Mutual Information between views/modalities should be considered
- Contrastive Methods could be leveraged with/instead of Predictive Methods[6]
- ...

[6] Contrastive Learning for Unpaired Image-to-Image Translation (ECCV 2020)

# Conclusions & Inspirations

- CMC enables the learning of unsupervised representations from **multiple views** of datasets
- Mutual Information between views/modalities should be considered
- Contrastive Methods could be leveraged with/instead of Predictive Methods[6]
- ...

[6] Contrastive Learning for Unpaired Image-to-Image Translation (ECCV 2020)

# Thank you!