

# Closed-Loop Data Transcription via Minimizing Rate Reduction

**Yi Ma**

EECS Department, UC Berkeley

January 28, 2022

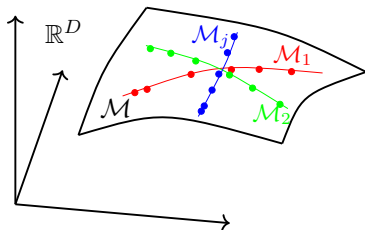
**Xili Dai, Shengbang Tong, Mingyang Li, Ziyang Wu, Michael Psenka,  
Bill Zhai, Yaodong Yu, Kwan Ho Ryan Chan, Xiaojun Yuan, Harry Shum**



# Outline

- 1 Motivation: Objectives of Learning from Data
- 2 LDR Representation via Principle of Rate Reduction
  - Theoretical justification
  - Experimental results
- 3 Transcription: Close the Loop of Encoding and Decoding
  - A closed-Loop formulation
  - Empirical verification
- 4 Conclusions and Open Problems

# High-Dim Data with Mixed Low-Dim Structures

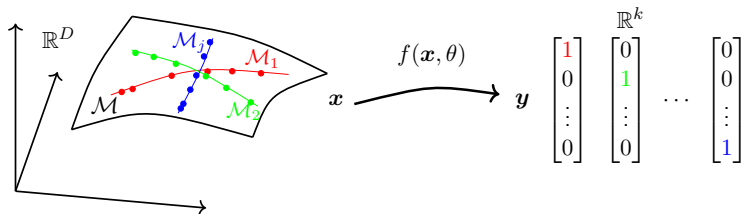


**Figure: High-dimensional Real-World Data:**  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]$  in  $\mathbb{R}^D$  lying on a mixture of low-dimensional submanifolds  $\cup_{j=1}^k \mathcal{M}_j \subset \mathbb{R}^D$ .

The main objective of learning from (samples of) real-world data:

**Find a most compact and simple representation of the data.**

# Fitting Class Labels via a Deep Network



**Figure: Black Box Classification:**  $y$  is the class label of  $x$  represented as a “one-hot” vector in  $\mathbb{R}^k$ . To learn a nonlinear mapping  $f(\cdot, \theta) : x \mapsto y$ , say modeled by a deep network, using cross-entropy (CE) loss.

$$\min_{\theta \in \Theta} \text{CE}(\theta, \mathbf{x}, \mathbf{y}) \doteq -\mathbb{E}[\langle \mathbf{y}, \log[f(\mathbf{x}, \theta)] \rangle] \approx -\frac{1}{m} \sum_{i=1}^m \langle \mathbf{y}_i, \log[f(\mathbf{x}_i, \theta)] \rangle. \quad (1)$$

*Prevalence of neural collapse during the terminal phase of deep learning training,*  
Papayan, Han, and Donoho, 2020.

# Represent Multi-class Multi-dimensional Data

Given samples

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m] \subset \mathbb{R}^D$$

from a mixture of

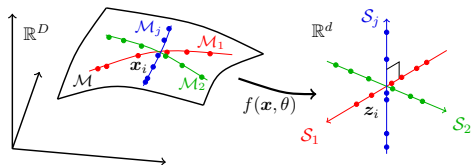
$k$  submanifolds:  $\mathcal{M} = \cup_{j=1}^k \mathcal{M}_j$ ,

**seek a good representation**

$\mathbf{Z} = [z_1, \dots, z_m] \subset \mathbb{R}^d$  through

a continuous mapping:

$$f(\mathbf{x}, \theta) : \mathbf{x} \in \mathbb{R}^D \mapsto \mathbf{z} \in \mathbb{R}^d.$$



Goals of “**re-present**” the data:

- from non-parametric (samples) to more compact (models).
- from nonlinear structures in  $\mathbf{X}$  to linear in  $\mathbf{Z} \subset \cup_{j=1}^k \mathcal{S}_j$ .
- from separable  $\mathbf{X}$  to maximally discriminative  $\mathbf{Z}$ .

**What constitutes a good representation?** (why a DNN?)

# Seeking a Linear Discriminative Representation (LDR)

**Desiderata:** Representation  $z = f(x, \theta)$  have the following properties:

- 1 *Within-Class Compressible:* Features of the same class/cluster should be highly compressed in a **low-dimensional** linear subspace.
- 2 *Between-Class Discriminative:* Features of different classes/clusters should be in highly **incoherent** linear subspaces.
- 3 *Maximally Informative Representation:* Dimension (or variance) of features for each class/cluster should be **as large as possible**.

**Is there a principled objective for all such properties, together?**

# Compactness Measure for Linear/Gaussian Representation

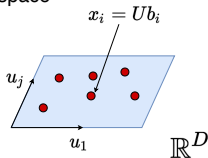
## Theorem (Coding Length, Ma, TPAMI'07)

The number of bits needed to encode data  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m] \in \mathbb{R}^{D \times m}$  up to a precision  $\|\mathbf{x} - \hat{\mathbf{x}}\|_2 \leq \epsilon$  is bounded by:

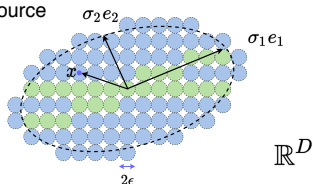
$$L(\mathbf{X}, \epsilon) \doteq \left( \frac{m + D}{2} \right) \log \det \left( \mathbf{I} + \frac{D}{m\epsilon^2} \mathbf{X} \mathbf{X}^\top \right).$$

This can be derived from constructively quantifying SVD of  $\mathbf{X}$  or by sphere packing  $\text{vol}(\mathbf{X})$  as samples of a noisy Gaussian source.

Linear subspace



Gaussian source



# Compactness Measure for Linear/Gaussian Representation

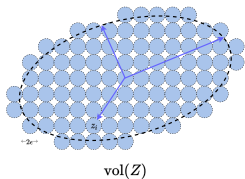
If  $\mathbf{X}$  is not (piecewise) linear or Gaussian, consider a **nonlinear** mapping:

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m] \in \mathbb{R}^{D \times m} \xrightarrow{f(\mathbf{x}, \theta)} \mathbf{Z}(\theta) = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m] \in \mathbb{R}^{d \times m}.$$

The average coding length per sample (rate) subject to a distortion  $\epsilon$ :

$$R(\mathbf{Z}, \epsilon) \doteq \frac{1}{2} \log \det \left( \mathbf{I} + \frac{d}{m\epsilon^2} \mathbf{Z} \mathbf{Z}^\top \right). \quad (2)$$

**Rate distortion is an intrinsic measure for the volume of all features.**





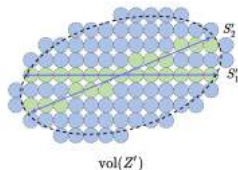
# Compactness Measure for Mixed Linear Representations

The features  $\mathbf{Z}$  of multi-class data

$$\mathbf{X} = \mathbf{X}_1 \cup \mathbf{X}_2 \cup \cdots \cup \mathbf{X}_k \subset \cup_{j=1}^k \mathcal{M}_j.$$

may be partitioned into multiple subsets:

$$\mathbf{Z} = \mathbf{Z}_1 \cup \mathbf{Z}_2 \cup \cdots \cup \mathbf{Z}_k \subset \cup_{j=1}^k \mathcal{S}_j.$$



W.r.t. this partition, the **average coding rate** is:

$$R^c(\mathbf{Z}, \epsilon | \mathbf{\Pi}) \doteq \sum_{j=1}^k \frac{\text{tr}(\mathbf{\Pi}_j)}{2m} \log \det \left( \mathbf{I} + \frac{d}{\text{tr}(\mathbf{\Pi}_j)\epsilon^2} \mathbf{Z} \mathbf{\Pi}_j \mathbf{Z}^\top \right), \quad (3)$$

where  $\mathbf{\Pi} = \{\mathbf{\Pi}_j \in \mathbb{R}^{m \times m}\}_{j=1}^k$  encode the membership of the  $m$  samples in the  $k$  classes: the diagonal entry  $\mathbf{\Pi}_j(i, i)$  of  $\mathbf{\Pi}_j$  is the probability of sample  $i$  belonging to subset  $j$ .  $\Omega \doteq \{\mathbf{\Pi} \mid \sum \mathbf{\Pi}_j = \mathbf{I}, \mathbf{\Pi}_j \geq \mathbf{0}\}$

# Measure for Linear Discriminative Representation (LDR)

**A Fundamental Idea:** maximize the **difference** between the coding rate of all features and the average rate of features in each of the classes:

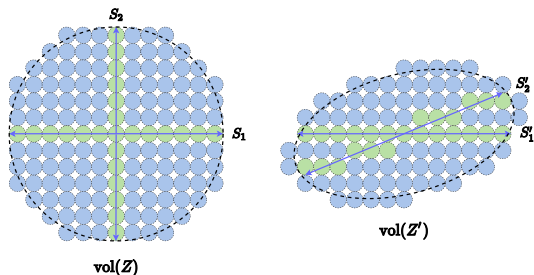
$$\Delta R(\mathbf{Z}, \mathbf{\Pi}, \epsilon) = \underbrace{\frac{1}{2} \log \det \left( \mathbf{I} + \frac{d}{m\epsilon^2} \mathbf{Z}\mathbf{Z}^\top \right)}_R - \sum_{j=1}^K \underbrace{\frac{\text{tr}(\mathbf{\Pi}_j)}{2m} \log \det \left( \mathbf{I} + \frac{d}{\text{tr}(\mathbf{\Pi}_j)\epsilon^2} \mathbf{Z}\mathbf{\Pi}_j\mathbf{Z}^\top \right)}_{R^c}.$$

This difference is called **rate reduction**:

- Large  $R$ : **expand** all features  $\mathbf{Z}$  as **large** as possible.
- Small  $R^c$ : **compress** each class  $\mathbf{Z}_j$  as **small** as possible.

**Slogan: similarity contracts and dissimilarity contrasts!**

# Interpretation of MCR<sup>2</sup>: Sphere Packing and Counting



**Example:** two subspaces  $S_1$  and  $S_2$  in  $\mathbb{R}^2$ .

- $\log \#(\text{green spheres} + \text{blue spheres}) = \text{rate of span of all samples } R.$
- $\log \#(\text{green spheres}) = \text{rate of the two subspaces } R^c.$
- $\log \#(\text{blue spheres}) = \text{rate reduction gain } \Delta R.$

# Principle of Maximal Coding Rate Reduction (MCR<sup>2</sup>)

[Yu, Chan, You, Song, Ma, NeurIPS2020]

Learn a mapping  $f(\mathbf{x}, \theta)$  (for a given partition  $\mathbf{\Pi}$ ):

$$\mathbf{X} \xrightarrow{f(\mathbf{x}, \theta)} \mathbf{Z}(\theta) \xrightarrow{\mathbf{\Pi}, \epsilon} \Delta R(\mathbf{Z}(\theta), \mathbf{\Pi}, \epsilon) \quad (4)$$

so as to **Maximize the Coding Rate Reduction (MCR<sup>2</sup>)**:

$$\begin{aligned} \max_{\theta} \quad & \Delta R(\mathbf{Z}(\theta), \mathbf{\Pi}, \epsilon) = R(\mathbf{Z}(\theta), \epsilon) - R^c(\mathbf{Z}(\theta), \epsilon \mid \mathbf{\Pi}), \\ \text{subject to} \quad & \|\mathbf{Z}_j(\theta)\|_F^2 = m_j, \mathbf{\Pi} \in \Omega. \end{aligned} \quad (5)$$

Since  $\Delta R$  is *monotonic* in the scale of  $\mathbf{Z}$ , one needs to:

**normalize the features**  $z = f(\mathbf{x}, \theta)$  **so as to compare**  $\mathbf{Z}(\theta)$  **and**  $\mathbf{Z}(\theta')$ !

Batch normalization, Sergey Ioffe and Christian Szegedy, 2015.

Layer normalization'16, instance normalization'16; group normalization'18...

# Theoretical Justification of the MCR<sup>2</sup> Principle

## Theorem (Informal Statement [Yu et.al., NeurIPS2020])

Suppose  $\mathbf{Z}^* = \mathbf{Z}_1^* \cup \dots \cup \mathbf{Z}_k^*$  is the optimal solution that maximizes the rate reduction (5). We have:

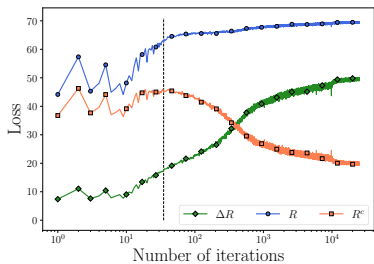
- *Between-class Discriminative: As long as the ambient space is adequately large ( $d \geq \sum_{j=1}^k d_j$ ), the subspaces are all orthogonal to each other, i.e.  $(\mathbf{Z}_i^*)^\top \mathbf{Z}_j^* = \mathbf{0}$  for  $i \neq j$ .*
- *Maximally Informative Representation: As long as the coding precision is adequately high, i.e.,  $\epsilon^4 < \min_j \left\{ \frac{m_j}{m} \frac{d^2}{d_j^2} \right\}$ , each subspace achieves its maximal dimension, i.e.  $\text{rank}(\mathbf{Z}_j^*) = d_j$ . In addition, the largest  $d_j - 1$  singular values of  $\mathbf{Z}_j^*$  are equal.*

A new slogan, beyond Aristotle:

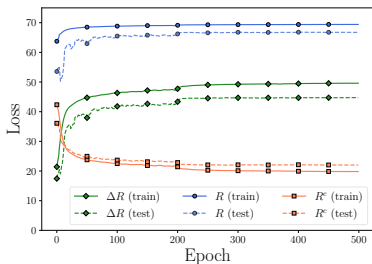
**The whole is to be maximally greater than the sum of the parts!**

# Experiment I: Supervised Deep Learning

**Experimental Setup:** Train  $f(x, \theta)$  as ResNet18 on the CIFAR10 dataset, feature  $z$  dimension  $d = 128$ , precision  $\epsilon^2 = 0.5$ .



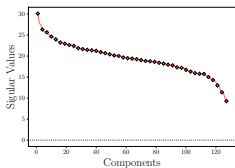
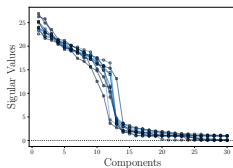
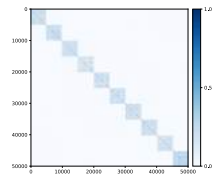
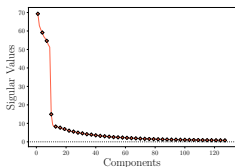
(a)



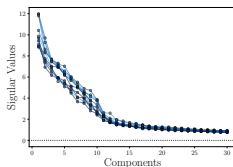
(b)

**Figure:** (a). Evolution of  $R, R^c, \Delta R$  during the training process; (b). Training loss versus testing loss.

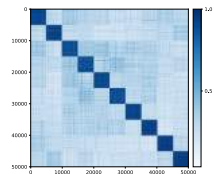
# Visualization of Learned Representations $\mathbb{Z}$

(a)  $MCR^2$  (overall)(b)  $MCR^2$  (PCA of every class)(c)  $MCR^2$  (cosine similarity)

(d) CE (overall)



(e) CE (PCA of every class)

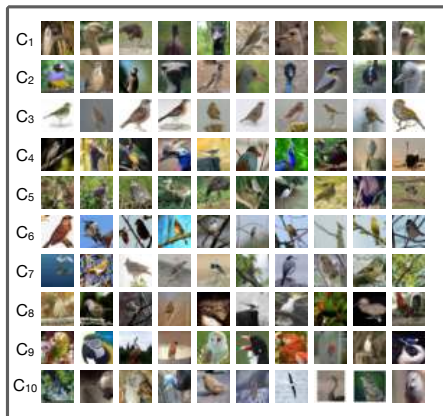


(f) CE (cosine similarity)

Figure: PCA of learned representations from  $MCR^2$  and cross-entropy.

**No neural collapse!**

# Visualization - Samples along Principal Components



(a) Bird



(b) Ship

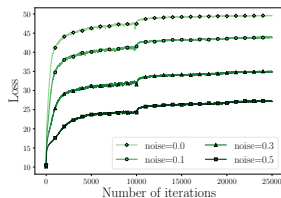
**Figure:** Top-10 “principal” images for class - “Bird” and “Ship” in the CIFAR10.



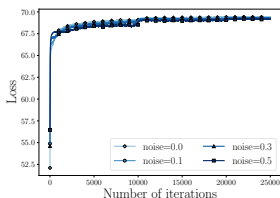
# Experiment II: Robustness to Label Noise

	RATIO=0.0	RATIO=0.1	RATIO=0.2	RATIO=0.3	RATIO=0.4	RATIO=0.5
CE TRAINING	0.939	0.909	0.861	0.791	0.724	0.603
MCR <sup>2</sup> TRAINING	<b>0.940</b>	<b>0.911</b>	<b>0.897</b>	<b>0.881</b>	<b>0.866</b>	<b>0.843</b>

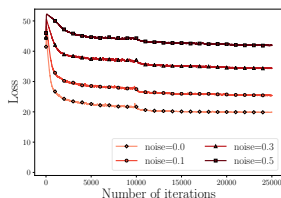
Table 1: Classification results with features learned with labels corrupted at different levels.



(a)  $\Delta R(\mathbf{Z}(\theta), \Pi, \epsilon)$



(b)  $R(\mathbf{Z}(\theta), \epsilon)$



(c)  $R^c(\mathbf{Z}(\theta), \epsilon | \Pi)$

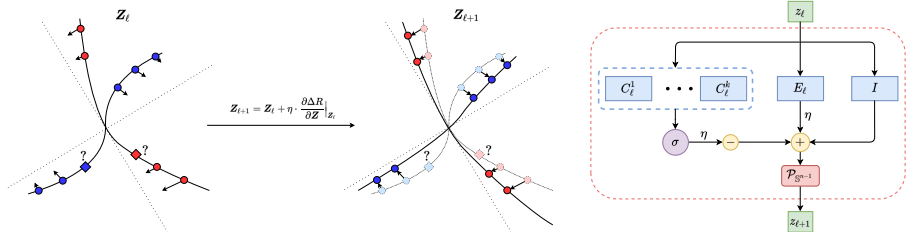
Figure: Evolution of  $R, R^c, \Delta R$  of MCR<sup>2</sup> during training with corrupted labels.

**Represent only what can be jointly compressed.**

# ReduNet: A White-box Deep Network from MCR<sup>2</sup>

A **white-box**, **forward-constructed**, **multi-channel convolution** deep neural network from maximizing the rate reduction via projected gradient flow:

$$\dot{\mathbf{Z}} = \eta \cdot \frac{\partial \Delta R}{\partial \mathbf{Z}} \quad \text{s.t.} \quad \mathbf{Z} \subset \mathbb{S}^{d-1}.$$

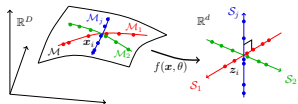


**ReduNet: A Whitebox Deep Network from Rate Reduction (JMLR'22):**

<https://arxiv.org/abs/2105.10446>

# Deep Networks for Linear Discriminative Representations

Comparison with conventional DNNs:



	Conventional DNNs	ReduNets
Objectives	label fitting	rate reduction
Deep architectures	trial & error	iterative optimization
Layer operators	empirical	projected gradient
Shift invariance	CNNs+augmentation	invariant ReduNets
Initializations	random/pre-design	forward computed
Training/fine-tuning	back prop	forward/back prop
Interpretability	black box	white box
Representations	hidden/latent	incoherent subspaces

## From One-sided to Closed-Loop Representation

$$\text{MCR}^2 : \mathbf{X} \xrightarrow{f(\mathbf{x}, \theta)} \mathbf{Z}(\theta) : \max_{\theta} \Delta R(\mathbf{Z}(\theta), \mathbf{\Pi}, \epsilon).$$

Features learned are more interpretable, independent, rich, and robust.

**However:**

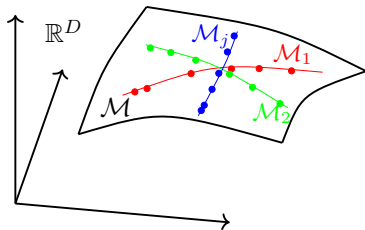
- Need to choose a proper feature dimension  $d$ .
- How good are the learned representation  $\mathbf{Z}$ ?
- Anything missing, anything unexpected:  $\dim(\mathbf{X}) = \dim(\mathbf{Z})$ ?
- Can we go from the feature  $\mathbf{Z}$  back to the data  $\mathbf{X}$ ?
- Is an LDR adequate to **generate** real-world (visual) data?

**Can we find a closed-loop (auto-encoding) data representation:**

$$\mathbf{X} \xrightarrow{f(\mathbf{x}, \theta)} \mathbf{Z}(\theta) \xrightarrow{g(\mathbf{z}, \eta)} \hat{\mathbf{X}}? \quad (6)$$

## Low-dim Representation for High-Dim Data

**Assumption:** the data  $\mathbf{X}$  lies on a low-dimensional submanifold  $\mathbf{X} \subset \mathcal{M}$  or multiple ones:  $\mathbf{X} \subset \cup_{j=1}^k \mathcal{M}_j$  in a high-dimensional space  $\in \mathbb{R}^D$ :



**Goal:** seeking a low-dim representation  $\mathbf{Z}$  in  $\mathbb{R}^d$  ( $d \ll D$ ) for the data  $\mathbf{X}$  on low-dim submanifolds such that:

$$\mathbf{X} \subset \mathbb{R}^D \xrightarrow{f(\mathbf{x}, \theta)} \mathbf{Z} \subset \mathbb{R}^d \xrightarrow{g(\mathbf{z}, \eta)} \hat{\mathbf{X}} \approx \mathbf{X} \in \mathbb{R}^D. \quad (7)$$

## Problem Formulation

**Desiderata** for a **good** representation:

- **Geometry:**  $f$  and  $g$  are continuous and **approximately isometric**.
- **Auto Encoding/Embedding** for the data  $\mathbf{X}$ :

$$g(f(\mathcal{M})) = \mathcal{M}, \quad \text{or} \quad g(f(\mathcal{M}_j)) = \mathcal{M}_j. \quad (8)$$

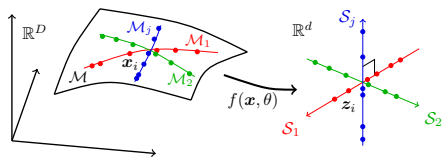
**Caveats:** we do not know  $\dim(\mathcal{M})$  nor  $d_j = \dim(\mathcal{M}_j)$ . Often

$$d > \dim(\mathcal{M}) \quad \text{or} \quad d > d_1 + d_2 + \cdots + d_k.$$

Structure of the learned  $\mathbf{Z} \subset f(\mathcal{M})$  often remains **“hidden”** in  $\mathbb{R}^d$ !

- **So further wish the feature  $\mathbf{Z}$  explicitly simple, say an LDR:**

$$\begin{aligned} f(\mathcal{M}) &= \mathcal{S} \quad \text{or} \\ f(\mathcal{M}_j) &= \mathcal{S}_j \quad (\text{with } \mathcal{S}_i \perp \mathcal{S}_j). \end{aligned}$$



## Three Classic Simpler Cases

**One low-dim linear subspace:** Principal Component Analysis (**PCA**)

$$\mathbf{X} \subset \mathcal{S}^D \xrightarrow{\mathbf{V}^T} \mathbf{Z} \subset \mathcal{S}^d \xrightarrow{\mathbf{V}} \hat{\mathbf{X}} \subset \mathcal{S}^D. \quad (9)$$

**Multiple linear subspaces:** Generalized PCA (**GPCA**)<sup>1</sup>

$$\mathbf{X} \subset \cup_{j=1}^k \mathcal{S}_j \xrightarrow{f(\mathbf{x}, \theta)} \cup_{j=1}^k \mathbf{Z}_j \subset \mathcal{S}_j \xrightarrow{g(\mathbf{z}, \eta)} \hat{\mathbf{X}} \subset \cup_{j=1}^k \mathcal{S}_j. \quad (10)$$

**One low-dim nonlinear submanifold:** **Nonlinear PCA**<sup>2</sup>

$$\mathbf{X} \subset \mathcal{M}^D \xrightarrow{f(\mathbf{x}, \theta)} \mathbf{Z} \subset \mathcal{S}^d \xrightarrow{g(\mathbf{z}, \eta)} \hat{\mathbf{X}} \subset \mathcal{M}^D. \quad (11)$$

**The most general, likely the most useful, case:**

$$\mathbf{X} \subset \cup_{j=1}^k \mathcal{M}_j \xrightarrow{f(\mathbf{x}, \theta)} \cup_{j=1}^k \mathbf{Z}_j \subset \mathcal{S}_j \xrightarrow{g(\mathbf{z}, \eta)} \hat{\mathbf{X}} \subset \cup_{j=1}^k \mathcal{M}_j. \quad (12)$$

<sup>1</sup>Generalized principal component analysis, R. Vidal, Yi Ma, and S. Sastry, 2005.

<sup>2</sup>Nonlinear PCA using autoassociative neural networks, M. Krammer, 1991.

# Principal Component Analysis (Auto Encoding)

**One low-dim linear subspace:** principal component analysis (PCA)

$$\mathbf{X} \subset \mathcal{S}^D \xrightarrow{\mathbf{V}^T} \mathbf{Z} \subset \mathcal{S}^d \xrightarrow{\mathbf{V}} \hat{\mathbf{X}} \subset \mathcal{S}^D. \quad (13)$$

Solve the following optimization problem:

$$\min_{\mathbf{V}} \|\mathbf{X} - \hat{\mathbf{X}}\|_2^2 \quad \text{s.t.} \quad \hat{\mathbf{X}} = \mathbf{V}\mathbf{V}^T\mathbf{X}, \quad \mathbf{V} \in \mathbf{O}(D, d). \quad (14)$$



# Principal Component Analysis (Auto Encoding)

**One low-dim linear subspace:** principal component analysis (PCA)

$$\mathbf{X} \subset \mathcal{S}^D \xrightarrow{\mathbf{V}^T} \mathbf{Z} \subset \mathcal{S}^d \xrightarrow{\mathbf{V}} \hat{\mathbf{X}} \subset \mathcal{S}^D. \quad (13)$$

Solve the following optimization problem:

$$\min_{\mathbf{V}} \|\mathbf{X} - \hat{\mathbf{X}}\|_2^2 \quad \text{s.t.} \quad \hat{\mathbf{X}} = \mathbf{V}\mathbf{V}^T\mathbf{X}, \quad \mathbf{V} \in \mathbf{O}(D, d). \quad (14)$$

**One low-dim nonlinear submanifold:** Nonlinear PCA

$$\mathbf{X} \subset \mathcal{M}^D \xrightarrow{f(\mathbf{x}, \theta)} \mathbf{Z} \subset \mathcal{S}^d \xrightarrow{g(\mathbf{z}, \eta)} \hat{\mathbf{X}} \subset \mathcal{M}^D. \quad (15)$$

Solve the following optimization problem:

$$\min_{\theta, \eta} \underbrace{\|\mathbf{X} - \hat{\mathbf{X}}\|_2^2}_{d(\mathbf{X}, \hat{\mathbf{X}})^2} \quad \text{s.t.} \quad \hat{\mathbf{X}} = g(f(\mathbf{X}, \eta), \theta). \quad (16)$$

**What is the right distance  $d(\mathbf{X}, \hat{\mathbf{X}})$ , say for images?**

## Auto Encoding and its Difficulties

Nonlinear PCA: Auto-encoding (AE) (Krammer'91)

$$\mathbf{X} \subset \mathcal{M}^D \xrightarrow{f(\mathbf{x}, \theta)} \mathbf{Z} \subset \mathcal{S}^d \xrightarrow{g(\mathbf{z}, \eta)} \hat{\mathbf{X}} \subset \mathcal{M}^D. \quad (17)$$

Assuming a **generative** model:  $p(\mathbf{x}|\mathbf{z}, \Theta)$  and  $p(\mathbf{z}, \Theta)$ , **maximal likelihood**:


$$\max_{\Theta} P(\mathbf{X}, \Theta) \sim p(\mathbf{x}, \Theta) = \int p(\mathbf{x}|\mathbf{z}, \Theta)p(\mathbf{z}, \Theta)d\mathbf{z}. \quad (18)$$

is in general **intractable**, so is to compute the true posterior

$$P(\mathbf{Z}|\mathbf{X}, \Theta) \sim p(\mathbf{z}|\mathbf{x}, \Theta) = p(\mathbf{x}|\mathbf{z}, \Theta)p(\mathbf{z}, \Theta)/p(\mathbf{x}, \Theta). \quad (19)$$

Instead optimize certain **variational lower bounds** (VAE):<sup>3</sup>

$$\max -\mathcal{D}_{KL}(\underbrace{\hat{p}(\mathbf{z}|\mathbf{x}, \eta)}_{\text{surrogate}}, p(\mathbf{z}, \Theta)) + \mathbb{E}_{\hat{p}(\mathbf{z}|\mathbf{x}, \eta)}[\log p(\mathbf{x}|\mathbf{z}, \Theta)]. \quad (20)$$

<sup>3</sup>Auto-Encoding Variational Bayes, D. Kingma and M. Welling, 2014. 

## GAN and its Caveats

Learning generative models via **discriminative** approaches? (Tu'2007)

Generative Adversarial Nets (GAN) (Goodfellow'2014):

$$\mathbf{Z} \xrightarrow{g(\mathbf{z}, \eta)} \hat{\mathbf{X}}, \mathbf{X} \xrightarrow{d(\mathbf{x}, \theta)} \mathbf{0}, \mathbf{1}. \quad (21)$$

A **minimax game** between generator and discriminator:

$$\min_{\eta} \max_{\theta} \mathbb{E}_{p(\mathbf{x})} [\log d(\mathbf{x}, \theta)] + \mathbb{E}_{p(\mathbf{z})} [1 - \log d(\underbrace{g(\mathbf{z}, \eta), \theta}_{\hat{\mathbf{x}} \sim p_g})]. \quad (22)$$

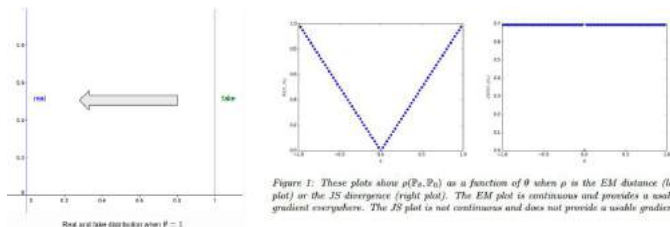
This is equivalent to minimize the *Jensen-Shannon divergence*:

$$\mathcal{D}_{JS}(p, p_g) = \mathcal{D}_{KL}(p \parallel (p + p_g)/2) + \mathcal{D}_{KL}(p_g \parallel (p + p_g)/2). \quad (23)$$

**But the J-S divergence is extremely difficult,  
if not impossible, to compute and optimize.**

# GAN and its Caveats

**An Example:** distance between distributions in high-dim space with non-overlapping low-dim supports. (always the case in high-dim!)



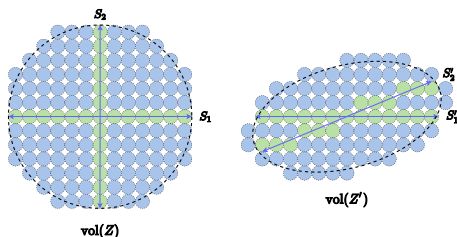
Replace  $\mathcal{D}_{JS}$  with the *Earth-Mover* distance or *Wasserstein-1 distance*:

$$W_1(p, p_g) = \inf_{\pi \in \Pi(p, p_g)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \pi} [\|\mathbf{x} - \mathbf{y}\|_1]. \quad (24)$$

- **Hard to compute**  $\mathcal{D}_{JS}(p, p_g)$  or  $W_1(p, p_g)$  accurately and efficiently.
- Either  $\mathcal{D}_{JS}$  or  $W_1$  has **no closed-form** even between two Gaussians!

# Rate Reduction as Distance between Subspace Gaussians

Rate reduction  $\Delta R = \log \#(\text{blue spheres})$  gives a **closed-form distance** between two (non-overlapping) subspace Gaussians  $S_1$  and  $S_2$ !



A good measure for the (LDR-like) features  $Z$ , but what about  $d(\mathbf{X}, \hat{\mathbf{X}})$ ?

$$\mathbf{X} \xrightarrow{f(\mathbf{x}, \theta)} \mathbf{Z} \xrightarrow{g(\mathbf{z}, \eta)} \hat{\mathbf{X}}. \quad (25)$$

**Question: do we ever need to measure in the data  $x$  space?**

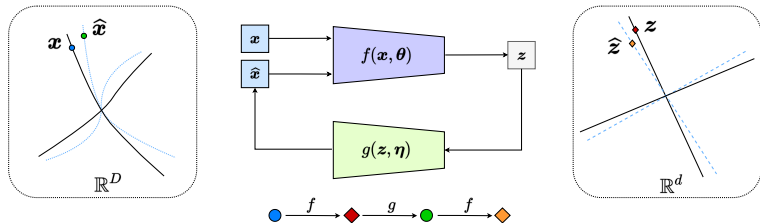
# A New Closed-Loop Formulation

**Goal:** Transcribe the data  $\mathbf{X} \subset \cup_{j=1}^k \mathcal{M}_j$  onto **an LDR**  $\mathbf{Z} \subset \cup_{j=1}^k \mathcal{S}_j$ :

$$\underbrace{f(\mathcal{M}_j) = \mathcal{S}_j}_{\text{linear}} \quad \text{with} \quad \underbrace{\mathcal{S}_i \perp \mathcal{S}_j}_{\text{discriminative}} \quad \text{and} \quad \underbrace{g(f(\mathcal{M}_j)) = \mathcal{M}_j}_{\text{auto-embedding}}. \quad (26)$$

Is it possible to measure everything internally in the feature space?

$$\mathbf{X} \xrightarrow{f(x,\theta)} \mathbf{Z} \xrightarrow{g(z,\eta)} \hat{\mathbf{X}} \xrightarrow{f(x,\theta)} \hat{\mathbf{Z}}. \quad (27)$$



## Measure Data Difference through Their Features

Measure difference in  $\mathbf{X}_j$  and  $\hat{\mathbf{X}}_j$  through their features  $\mathbf{Z}_j$  and  $\hat{\mathbf{Z}}_j$ :

$$\mathbf{X}_j \xrightarrow{f(\mathbf{x},\theta)} \mathbf{Z}_j \xrightarrow{g(\mathbf{z},\eta)} \hat{\mathbf{X}}_j \xrightarrow{f(\mathbf{x},\theta)} \hat{\mathbf{Z}}_j, \quad j = 1, \dots, k. \quad (28)$$

with the rate reduction measuring the error:

$$\Delta R(\mathbf{Z}_j, \hat{\mathbf{Z}}_j) \doteq R(\mathbf{Z}_j \cup \hat{\mathbf{Z}}_j) - \frac{1}{2}(R(\mathbf{Z}_j) + R(\hat{\mathbf{Z}}_j)), \quad j = 1, \dots, k. \quad (29)$$

## Measure Data Difference through Their Features

Measure difference in  $\mathbf{X}_j$  and  $\hat{\mathbf{X}}_j$  through their features  $\mathbf{Z}_j$  and  $\hat{\mathbf{Z}}_j$ :

$$\mathbf{X}_j \xrightarrow{f(\mathbf{x},\theta)} \mathbf{Z}_j \xrightarrow{g(\mathbf{z},\eta)} \hat{\mathbf{X}}_j \xrightarrow{f(\mathbf{x},\theta)} \hat{\mathbf{Z}}_j, \quad j = 1, \dots, k. \quad (28)$$

with the rate reduction measuring the error:

$$\Delta R(\mathbf{Z}_j, \hat{\mathbf{Z}}_j) \doteq R(\mathbf{Z}_j \cup \hat{\mathbf{Z}}_j) - \frac{1}{2}(R(\mathbf{Z}_j) + R(\hat{\mathbf{Z}}_j)), \quad j = 1, \dots, k. \quad (29)$$

**Decoder/controller**  $g$  **minimizes** the difference between  $\mathbf{X}$  and  $\hat{\mathbf{X}}$ :

$$d(\mathbf{X}, \hat{\mathbf{X}}) \doteq \min_{\eta} \sum_{j=1}^k \Delta R(\mathbf{Z}_j, \hat{\mathbf{Z}}_j) = \min_{\eta} \sum_{j=1}^k \Delta R(\mathbf{Z}_j, f(g(\mathbf{Z}_j, \eta), \theta)).$$



## Measure Data Difference through Their Features

Measure difference in  $\mathbf{X}_j$  and  $\hat{\mathbf{X}}_j$  through their features  $\mathbf{Z}_j$  and  $\hat{\mathbf{Z}}_j$ :

$$\mathbf{X}_j \xrightarrow{f(\mathbf{x}, \theta)} \mathbf{Z}_j \xrightarrow{g(\mathbf{z}, \eta)} \hat{\mathbf{X}}_j \xrightarrow{f(\mathbf{x}, \theta)} \hat{\mathbf{Z}}_j, \quad j = 1, \dots, k. \quad (28)$$

with the rate reduction measuring the error:

$$\Delta R(\mathbf{Z}_j, \hat{\mathbf{Z}}_j) \doteq R(\mathbf{Z}_j \cup \hat{\mathbf{Z}}_j) - \frac{1}{2}(R(\mathbf{Z}_j) + R(\hat{\mathbf{Z}}_j)), \quad j = 1, \dots, k. \quad (29)$$

**Decoder/controller**  $g$  **minimizes** the difference between  $\mathbf{X}$  and  $\hat{\mathbf{X}}$ :

$$d(\mathbf{X}, \hat{\mathbf{X}}) \doteq \min_{\eta} \sum_{j=1}^k \Delta R(\mathbf{Z}_j, \hat{\mathbf{Z}}_j) = \min_{\eta} \sum_{j=1}^k \Delta R(\mathbf{Z}_j, f(g(\mathbf{Z}_j, \eta), \theta)).$$

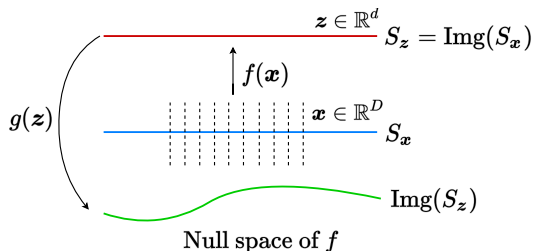
**Encoder/sensor**  $f$  **amplifies** any difference between  $\mathbf{X}$  and  $\hat{\mathbf{X}}$ :

$$d(\mathbf{X}, \hat{\mathbf{X}}) \doteq \max_{\theta} \sum_{j=1}^k \Delta R(\mathbf{Z}_j, \hat{\mathbf{Z}}_j) = \max_{\theta} \sum_{j=1}^k \Delta R(f(\mathbf{X}_j, \theta), f(\hat{\mathbf{X}}_j, \theta)).$$

## Dual Roles of the Encoder and Decoder

The encoder  $f$  needs to be a **discriminative sensor** that can discern and amplify any error between the distributions between  $\mathbf{X}$  and  $\hat{\mathbf{X}}$ .

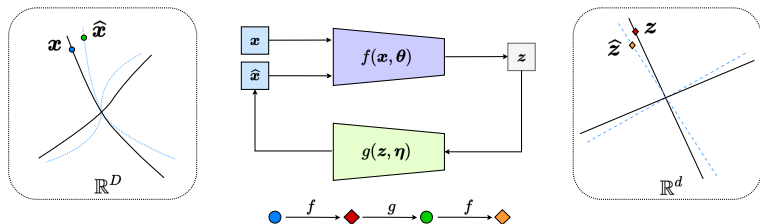
**Reason:** for a fixed encoder  $f$ , the decoder  $g$  can easily produce an ambiguous decoding such that the error between  $\mathbf{Z}$  and  $\hat{\mathbf{Z}}$  is zero!



$$g \circ f \neq \text{Id}, \text{ but } f \circ g = \text{Id}$$

## Dual Roles of the Encoder and Decoder

$f$  is both an encoder and sensor; and  $g$  is both a decoder and controller. They form a **closed-loop feedback control system**:



A closed-loop notion of “self-consistency” between  $\mathbf{X}$  and  $\hat{\mathbf{X}}$  is given by a **pursuit-evasion game** between  $f$  as a “evader” and  $g$  as a “pursuer”:

$$\mathcal{D}(\mathbf{X}, \hat{\mathbf{X}}) \doteq \min_{\eta} \max_{\theta} \sum_{j=1}^k \Delta R \left( \underbrace{f(\mathbf{X}_j, \theta)}_{\mathbf{Z}_j(\theta)}, \underbrace{f(g(f(\mathbf{X}_j, \theta), \eta), \theta)}_{\hat{\mathbf{Z}}_j(\theta, \eta)} \right). \quad (30)$$

## Overall Objective: Self-Consistency & Parsimony

The overall **minimax game** between the encoder  $f$  and decoder  $g$ :

- $f$  *maximizes* the rate reduction of the features  $\mathbf{Z}$  of the data  $\mathbf{X}$ ;
- $g$  *minimizes* the rate reduction of the features  $\hat{\mathbf{Z}}$  of the decoded  $\hat{\mathbf{X}}$ .

A minimax program to learn a **multi-class LDR** for data  $\mathbf{X} = \cup_{j=1}^k \mathbf{X}_j$ :

$$\min_{\eta} \max_{\theta} \underbrace{\Delta R(f(\mathbf{X}, \theta))}_{\text{Expansive encode}} + \underbrace{\Delta R(h(\mathbf{X}, \theta, \eta))}_{\text{Compressive decode}} + \sum_{j=1}^k \underbrace{\Delta R(f(\mathbf{X}_j, \theta), h(\mathbf{X}_j, \theta, \eta))}_{\text{Contrastive \& Contractive}}$$

with  $h(\mathbf{x}) \doteq f \circ g \circ f(\mathbf{x})$ , or equivalently

$$\min_{\eta} \max_{\theta} \Delta R(\mathbf{Z}(\theta)) + \Delta R(\hat{\mathbf{Z}}(\theta, \eta)) + \sum_{j=1}^k \Delta R(\mathbf{Z}_j(\theta), \hat{\mathbf{Z}}_j(\theta, \eta)).$$

## Overall Objective: Self-Consistency & Parsimony

The overall **minimax game** between the encoder  $f$  and decoder  $g$ :

- $f$  *maximizes* the rate reduction of the features  $\mathbf{Z}$  of all the data  $\mathbf{X}$ ;
- $g$  *minimizes* the rate reduction of the features  $\hat{\mathbf{Z}}$  of the decoded  $\hat{\mathbf{X}}$ .

A minimax program to learn a **one-class LDR** for data  $\mathbf{X}$ :

$$\text{Binary: } \min_{\eta} \max_{\theta} \underbrace{\Delta R(f(\mathbf{X}, \theta), h(\mathbf{X}, \theta, \eta))}_{\text{Contrastive \& Contractive}}$$

or equivalently

$$\text{Binary: } \min_{\eta} \max_{\theta} \Delta R(\mathbf{Z}(\theta), \hat{\mathbf{Z}}(\theta, \eta)).$$

## Characteristics of the Overall Objective

$$\min_{\eta} \max_{\theta} \Delta R(\mathbf{Z}(\theta)) + \Delta R(\hat{\mathbf{Z}}(\theta, \eta)) + \sum_{j=1}^k \Delta R(\mathbf{Z}_j(\theta), \hat{\mathbf{Z}}_j(\theta, \eta)).$$

- **Simplicity:** all terms are **closed-form** rate reduction on features.
- **Consistency:** **closed-loop** encoding and decoding are all needed.
- **Explicit:** distribution of learned features  $\mathbf{Z}$  is **not hidden** (an LDR).
- **No** need of any direct explicit distance between  $\mathbf{X}$  and  $\hat{\mathbf{X}}$ .
- **No** need to specify a prior or a surrogate target distribution.
- **No** more approximations or bounds for (KL-, JS-, W-) “distances”.
- **No** heuristics or regularizing terms.

**Self-consistency and Parsimony are all you need to model  $\mathbf{X}$ ?**

# Empirical Verification on Visual Data

## Experimental Setup:

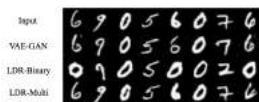
- **Datasets:** MNIST, CIFAR10, STL-10, CelebA faces, LSUN bedroom, ImageNet
- **Network architectures:** basic DCGAN & ResNet (**not customized**).
- **Feature space:** **the same** 128-dim regardless of data resolution or size
- **Quantization precision:** **the same**  $\epsilon^2 = 0.5$ .
- **Optimizer:** *Adam* with **the same** hyperparameters  $\beta_1 = 0, \beta_2 = 0.9$ .
- **Linear rate:** **the same** initial 0.00015 with linear decay.

**No other regularization, heuristics, or engineering tricks.**

# Empirical Verification: Fair Comparison to Baselines

Method		GAN	GAN (LDA-Binary)	VAE-GAN	LDA-Binary	LDA-Multi
MNIST	IS $\uparrow$	2.08	1.95	<b>2.21</b>	2.02	2.07
	FID $\downarrow$	24.78	20.15	33.65	<b>16.43</b>	16.47
CIFAR-10	IS $\uparrow$	7.32	7.23	7.11	<b>8.11</b>	7.13
	FID $\downarrow$	26.06	22.16	43.25	<b>19.63</b>	23.91

**Table:** Quantitative comparison on MNIST and CIFAR-10. Average Inception scores (IS) and FID scores.  $\uparrow$  means higher is better.  $\downarrow$  means lower is better.



(a) MNIST



(b) CIFAR-10



(c) ImageNet

**Figure:** Qualitative comparison on MNIST, CIFAR-10 and ImageNet.



# Empirical Verification on Visual Data

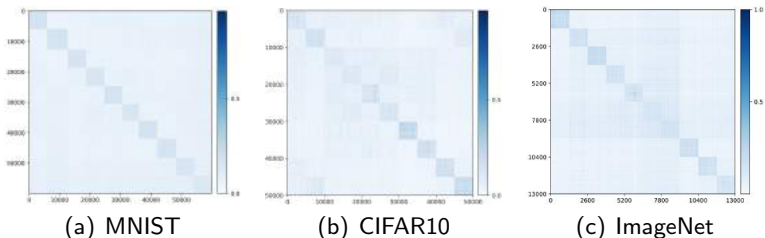


Figure: Visualizing the alignment between  $Z$  and  $\hat{Z}$ :  $|Z^\top \hat{Z}|$ .



Figure: Visualizing the auto-encoding property:  $x \approx \hat{x} = g \circ f(x)$ .

# Empirical Verification: Comparison on MNIST

(a) Original  $X$ (b) VAE-GAN  $\hat{X}$ (c) BiGAN  $\hat{X}$ (d) LDR-Binary  $\hat{X}$ (e) LDR-Multi  $\hat{X}$ 

**Figure:** Reconstruction results of different methods with the input data.

## Empirical Verification: MNIST PCAs

The feature  $z$  in each of the  $k$  principal subspaces can be modeled as a degenerate Gaussian from the PCA  $Z_j = V_j \Sigma_j U_j^T$ :

$$z_j \sim \bar{z}_j + \sum_{l=1}^{r_j} n_l^j \sigma_j^l v_j^l, \quad \text{where } n_l^j \sim \mathcal{N}(0, 1), \quad j = 1, \dots, k. \quad (31)$$



(a) ACGAN



(b) InfoGAN



(c) LDR-Multi

## Empirical Verification: MNIST PCAs

The feature  $z$  in each of the  $k$  principal subspaces can be modeled as a degenerate Gaussian from the PCA  $Z_j = V_j \Sigma_j U_j^T$ :

$$z_j \sim \bar{z}_j + \sum_{l=1}^{r_j} n_l^j \sigma_j^l v_j^l, \quad \text{where } n_l^j \sim \mathcal{N}(0, 1), \quad j = 1, \dots, k. \quad (32)$$

Nearest subspace classification based on the computed PCAs.

Table 3: Classification accuracy on MNIST, comparing to classifier based VAE methods (Parmar et al., 2021). Most of those VAE-based methods require auxiliary classifiers to boost classification performance.

Method	VAE	Factor VAE	Guide-VAE	DC-VAE	LDR-Binary	LDR-Multi
MNIST	97.12%	93.65%	98.51%	98.71%	89.12%	98.30%

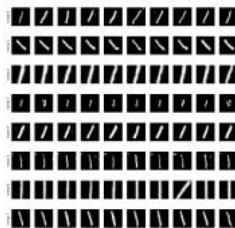
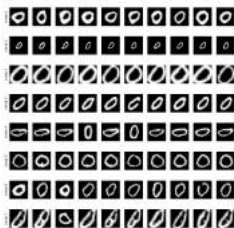
# Empirical Verification: Interpolation between Samples



Figure: Images generated from interpolating between samples in different classes.

# Empirical Verification: Transformed MNIST

Original data  $\mathbf{X}$  and their decoded version  $\hat{\mathbf{X}}$  on transformed MNIST.

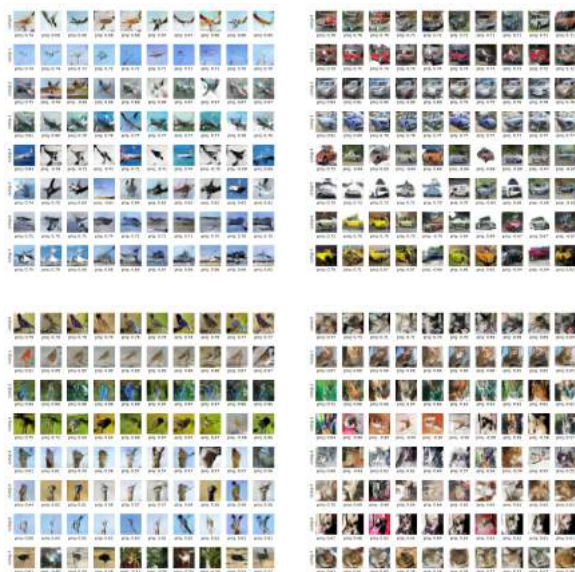


(c) Components of "0"

(d) Components of "1"

(e) Components of "2"

# Empirical Verification: “Principal Images” of CIFAR10

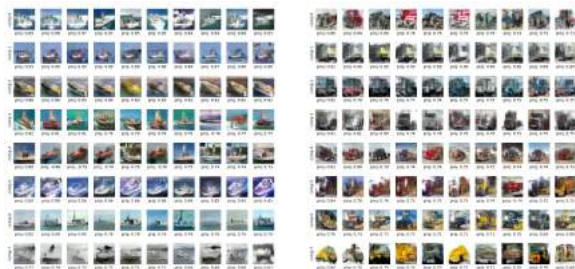


# Empirical Verification: “Principal Images” of CIFAR10





# Empirical Verification: “Principal Images” of CIFAR10

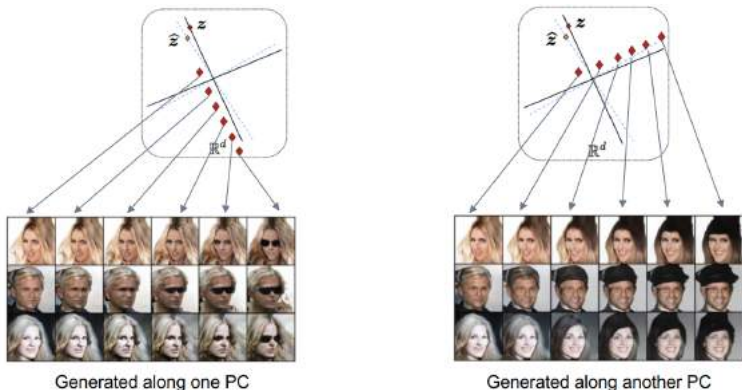


**Figure:** Reconstructed images  $\hat{X}$  from features  $Z$  close to the principal components learned for each of the 10 classes of CIFAR-10.

**Different classes are disentangled as principal subspaces.  
Visual attributes are disentangled as principal components.**

# Empirical Verification: Principal Components of CelebA

**Figure:** Generated images by sampling along the 9-th and 23-th principal components of the learned features  $Z$ , for the CelebA dataset.



**Visual attributes are disentangled as principal components.**

# Empirical Verification: CelebA Randomly Generated $\hat{X}$



# Empirical Verification: CelebA Input $X$



(a) Original  $X$

**Figure:** Visualizing the original  $x$  and corresponding decoded  $\hat{x}$  results on Celeb-A dataset. The LDR model is trained from LDR-Binary.

# Empirical Verification: CelebA Decoded $\hat{X}$



(a) Decoded  $\hat{X}$

**Figure:** Visualizing the original  $x$  and corresponding decoded  $\hat{x}$  results on Celeb-A dataset. The LDR model is trained from LDR-Binary.

# Empirical Verification: LSUN Bedroom Input $X$



(a) Original  $X$

**Figure:** Visualizing the original  $x$  and corresponding decoded  $\hat{x}$  results on LSUN-bedroom dataset. The LDR model is trained from LDR-Binary.

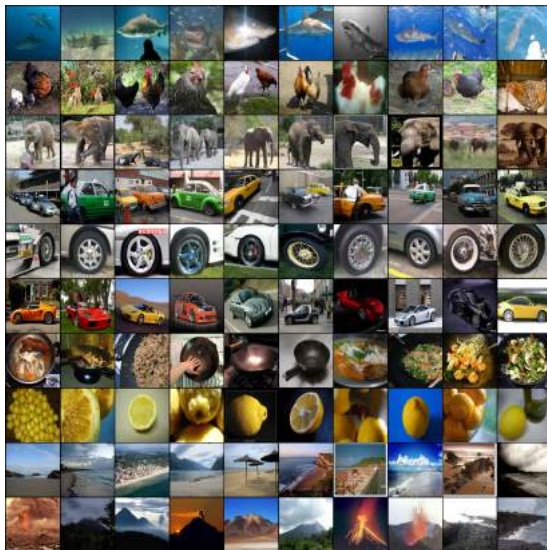
# Empirical Verification: LSUN Bedroom Decoded $\hat{X}$



(a) Decoded  $\hat{X}$

**Figure:** Visualizing the original  $x$  and corresponding decoded  $\hat{x}$  results on LSUN-bedroom dataset. The LDR model is trained from LDR-Binary.

# Empirical Verification: ImageNet 10-Class Input $X$

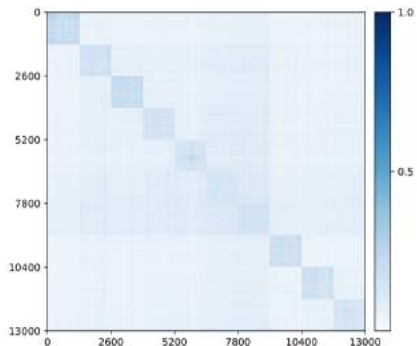


(a) Original  $X$

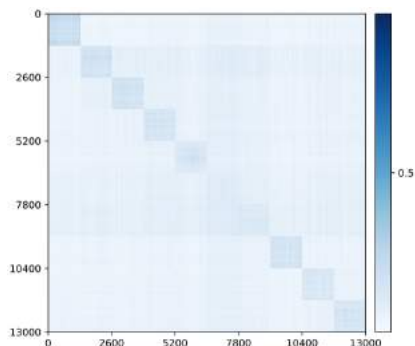


Empirical Verification: ImageNet 10-Class Decoded  $\hat{X}$ (b) Decoded  $\hat{X}$

# Empirical Verification: ImageNet Feature Similarity



(c)  $|Z^T Z|$



(d)  $|Z^T \hat{Z}|$

**Figure:** Visualizing feature alignment: (a) among features  $|Z^T Z|$ , (b) between features and decoded features  $|Z^T \hat{Z}|$ . These results obtained after 200,000 iterations.

# Empirical Verification: Quantitative

Table: Comparison on CIFAR-10, STL-10, and ImageNet.

Method	CIFAR-10		STL-10		ImageNet	
	IS $\uparrow$	FID $\downarrow$	IS $\uparrow$	FID $\downarrow$	IS $\uparrow$	FID $\downarrow$
<i>GAN based methods</i>						
DCGAN	6.6	-	7.8	-	-	-
SNGAN	7.4	29.3	<b>9.1</b>	40.1	-	48.73
CSGAN	8.1	19.6	-	-	-	-
LOGAN	<b>8.7</b>	<b>17.7</b>	-	-	-	-
<i>VAE/GAN based methods</i>						
VAE	3.8	115.8	-	-	-	-
VAE/GAN	7.4	39.8	-	-	-	-
NVAE	-	50.8	-	-	-	-
DC-VAE	<b>8.2</b>	<b>17.9</b>	8.1	41.9	-	-
LDR-Binary (ours)	<b>8.1</b>	<b>19.6</b>	8.4	<b>38.6</b>	7.74	<b>46.95</b>
LDR-Multi (ours)	7.1	23.9	7.7	45.7	6.44	55.51

# Empirical Verification: Ablation Study

Training the ImageNet with networks of different width.

	channel#=1024	channel#=512	channel#=256
BS=1800	success	success	success
BS=1600	success	success	success
BS=1024	failure	success	success
BS=800	failure	failure	success
BS=400	failure	failure	failure

**Table:** Ablation study on ImageNet about tradeoff between batch size (BS) and network width (channel #).

**No mode collapse!**

## Empirical Verification: Other Ablation Studies

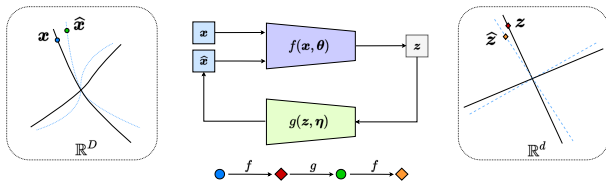
$$\min_{\eta} \max_{\theta} \Delta R(\mathbf{Z}(\theta)) + \Delta R(\hat{\mathbf{Z}}(\theta, \eta)) + \sum_{j=1}^k \Delta R(\mathbf{Z}_j(\theta), \hat{\mathbf{Z}}_j(\theta, \eta)).$$

Other ablations studies:

- the importance of the closed loop.
- the importance of rate reduction versus cross entropy.
- the three terms in the objective function.
- sensitivity to spectral normalization.
- choices in feature dimension or channel number.
- ...

see details in the paper <https://arxiv.org/abs/2111.06636>

# Conclusions: Closed-Loop Transcription to an LDR



- **universality:** embedding real-world data to a **simple and explicit** linear discriminative representation.
- **parsimony:** a **good tradeoff** in rate reduction via a minimax game between an encoder and a decoder.
- **feedback:** a **closed-loop feedback control** system between a sensor and a controller.
- **self-consistency:** **no need of any surrogate** distance in the external data space.

# Open Mathematical Problems

For the closed-loop minimax rate reduction program:

$$\min_{\eta} \max_{\theta} \Delta R(\mathbf{Z}(\theta)) + \Delta R(\hat{\mathbf{Z}}(\theta, \eta)) + \sum_{j=1}^k \Delta R(\mathbf{Z}_j(\theta), \hat{\mathbf{Z}}_j(\theta, \eta)).$$

- **optimality**: characterization of the **equilibrium points**?
- **convergence** of the closed-loop control problem (**infinite-dim**)?
- **linearization** of distribution supports (**plastic manifold learning**)?
- **optimal density** of the distributions (**Brascamp-Lieb inequalities**)?
- **guarantees** for approximate **sample-wise auto-encoding**?
- **correct model selection** (**no under- or over-fitting**)?

## Open Directions: Extensions and Connections

- How to **scale up** to hundreds and thousands of classes?  
(variational forms for rate reduction...)
- Internal computational mechanism for **memory** forming (in Nature)?  
(incremental learning without catastrophic forgetting...)
- Better **feedback** for generative quality and discriminative property?
- **Whitebox** architectures for closed-loop transcription (ReduNet like)?
- Closed-loop transcription to **other types of low-dim structures**?  
(dynamical, causal, logical, symbolical, graphical...)

The principles of **parsimony and self-consistency** shall always rule!



# References: Learning via Rate Reduction and Transcription

- 1 **Closed-Loop Data Transcription to an LDR via Minimizing Rate Reduction**  
<https://arxiv.org/abs/2111.06636> (under submission)
- 2 **ReduNet: A Whitebox Deep Network from Rate Reduction (JMLR'22):**  
<https://arxiv.org/abs/2105.10446>
- 3 **Representation** via Maximal Coding Rate Reduction (NeurIPS'20):  
<https://arxiv.org/abs/2006.08558>
- 4 **Classification** via Minimal Incremental Coding Length (NIPS 2007):  
[http://people.eecs.berkeley.edu/~yima/psfile/MICL\\_SJIS.pdf](http://people.eecs.berkeley.edu/~yima/psfile/MICL_SJIS.pdf)
- 5 **Clustering** via Lossy Coding and Compression (TPAMI 2007):  
<http://people.eecs.berkeley.edu/~yima/psfile/Ma-PAMI07.pdf>

**Parsimony and self-consistency** are all you need to  
learn  
a **compact and simple** memory for real-world data.

Thank you!  
Questions, please?

*“Learners need endless feedback more than they need endless teaching.”*

– Grant Wiggins

