Closed-Loop Data Transcription to an LDR via Minimaxing Rate Reduction
(Lecture 23)

Yi Ma

EECS Department, UC Berkeley

November 30, 2021

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

"Learners need endless feedback more than they need endless teaching."

– Grant Wiggins

High-Dim Data with Mixed Low-Dim Structures

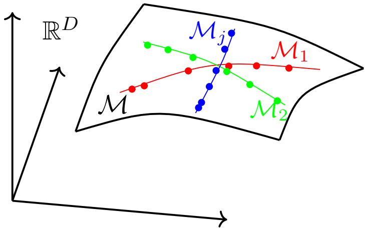


Figure: **High-dimensional Real-World Data**: $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m]$ in $\mathbb{R}^D$ lying on a mixture of low-dimensional submanifolds $\cup_{j=1}^{k} \mathcal{M}_j \subset \mathbb{R}^D$.

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

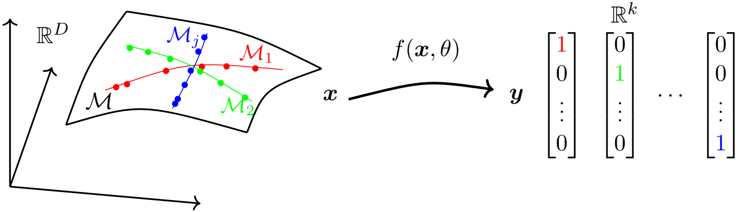


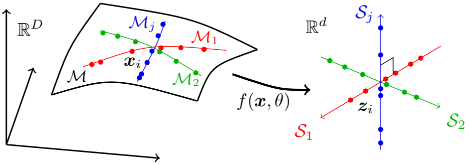
Figure: **Black Box Classification:** $\boldsymbol{y}$ is the class label of $\boldsymbol{x}$ represented as a "one-hot" vector in $\mathbb{R}^k$. To learn a nonlinear mapping $f(\cdot, \theta) : \boldsymbol{x} \mapsto \boldsymbol{y}$, say modeled by a deep network, using cross-entropy (CE) loss.

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

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

Represent Multi-class Multi-dimensional Data

Given samples $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m] \subset \mathbb{R}^D$ from a mixture of k submanifolds: $\mathcal{M} = \cup_{j=1}^{k} \mathcal{M}_j$, **seek a good representation** $\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m] \subset \mathbb{R}^d$ through a continuous mapping:



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

Goals of **“re-present”** the data:

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

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

Seeking a Linear Discriminative Representation (LDR)

Desiderata: Representation $\boldsymbol{z} = f(\boldsymbol{x}, \theta)$ 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 measure for all such properties, together?

Compactness Measure for Linear/Gaussian Representation

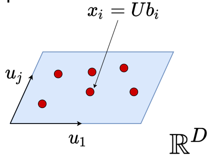
Theorem (Ma, TPAMI'07)

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

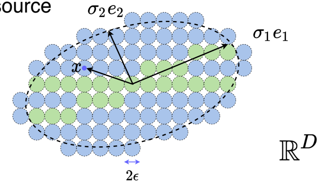
$$L(\boldsymbol{X}, \epsilon) \doteq \left(\frac{m+D}{2}\right) \log\det\left(\boldsymbol{I} + \frac{D}{m\epsilon^2}\boldsymbol{X}\boldsymbol{X}^\top\right).$$

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

Linear subspace



Gaussian source



Compactness Measure for Linear/Gaussian Representation

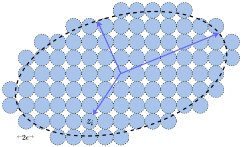
If $\boldsymbol{X}$ is not (piecewise) linear or Gaussian, consider a nonlinear mapping:

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

The average coding length per sample (rate) subject to a distortion ϵ:

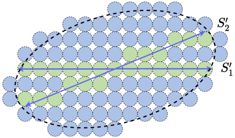
$$R(\boldsymbol{Z}, \epsilon) \doteq \frac{1}{2}\log\det\left(\boldsymbol{I} + \frac{d}{m\epsilon^2}\boldsymbol{Z}\boldsymbol{Z}^\top\right). \quad (2)$$

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



$\mathrm{vol}(Z)$

Compactness Measure for Mixed Linear Representations



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

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

may be partitioned into multiple subsets:

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

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

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

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

Measure for Linear Discriminative Representation (LDR)

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

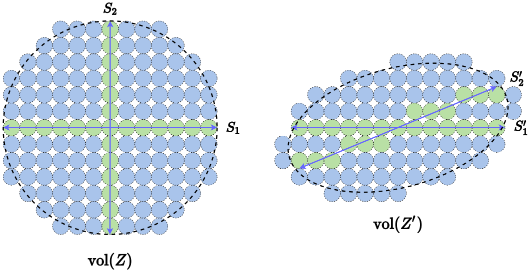
$$\Delta R(\boldsymbol{Z}, \boldsymbol{\Pi}, \epsilon) = \underbrace{\frac{1}{2} \log\det\left(\boldsymbol{I} + \frac{d}{m\epsilon^2}\boldsymbol{Z}\boldsymbol{Z}^\top\right)}_{R} - \underbrace{\sum_{j=1}^{k} \frac{\mathrm{tr}(\boldsymbol{\Pi}_j)}{2m} \log\det\left(\boldsymbol{I} + \frac{d}{\mathrm{tr}(\boldsymbol{\Pi}_j)\epsilon^2}\boldsymbol{Z}\boldsymbol{\Pi}_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

This difference is called **rate reduction**:

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

Slogan: similarity contracts and dissimilarity contrasts!

Interpretation of MCR^2: Sphere Packing and Counting



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

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

Principle of Maximal Coding Rate Reduction (MCR^2)

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

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

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

so as to **M**aximize the **C**oding **R**ate **R**eduction (**MCR**^{2}):

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

Since ΔR is *monotonic* in the scale of $\boldsymbol{Z}$, one needs to:
normalize the features $\boldsymbol{z} = f(\boldsymbol{x}, \theta)$ so as to compare $\boldsymbol{Z}(\theta)$ and $\boldsymbol{Z}(\theta')$!

Batch normalization, Sergey Ioffe and Christian Szegedy, 2015.

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

Theoretical Justification of the MCR^2 Principle

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

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

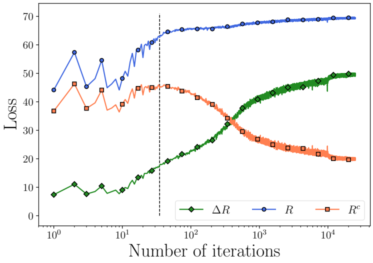
- Between-class Discriminative: *As long as the ambient space is adequately large* ($d \geq \sum_{j=1}^k d_j$)*, the subspaces are all orthogonal to each other,* i.e. $(\boldsymbol{Z}_i^\star)^\top \boldsymbol{Z}_j^\star = \mathbf{0}$ *for* $i \neq j$*.*
- Maximally Informative Representation: *As long 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.* $\operatorname{rank}(\boldsymbol{Z}_j^\star) = d_j$*. In addition, the largest* $d_j - 1$ *singular values of* $\boldsymbol{Z}_j^\star$ *are equal.*

A new slogan, beyond Aristotle:

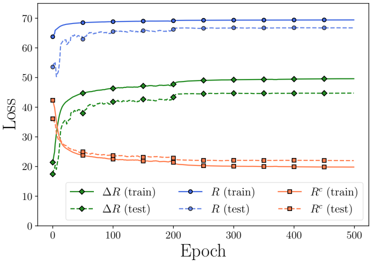
The whole is **to be maximally** *greater than the sum of the parts!*

Experiment I: Supervised Deep Learning

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



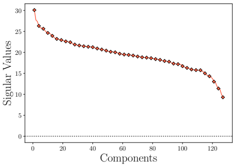
(a)



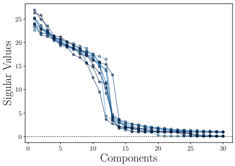
(b)

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

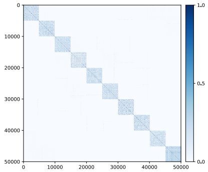
Visualization of Learned Representations Z



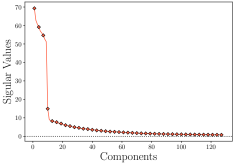
(a) MCR^2 (overall)



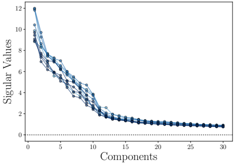
(b) MCR^2 (PCA of every class)



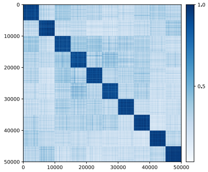
(c) MCR^2 (cosine similarity)



(d) CE (overall)



(e) CE (PCA of every class)



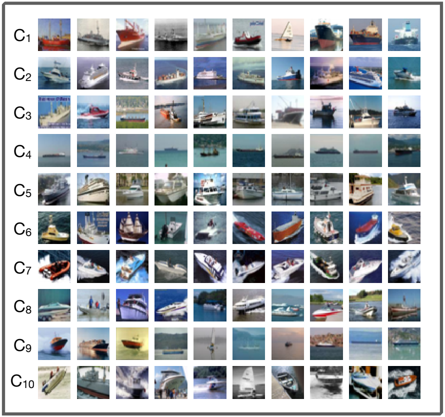
(f) CE (cosine similarity)

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

No neural collapse!

Visualization - Samples along Principal Components

(a) Bird



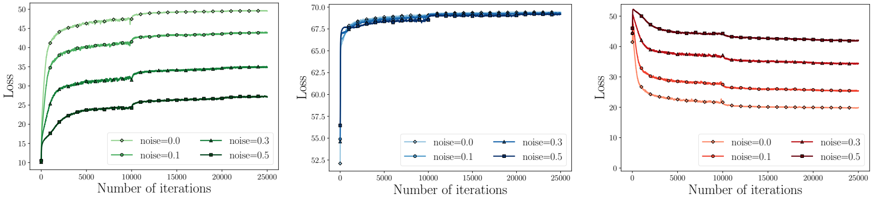
(b) Ship

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

Experiment II: Robustness to Label Noise

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

	RATIO=0.1	RATIO=0.2	RATIO=0.3	RATIO=0.4	RATIO=0.5
CE TRAINING	90.91%	86.12%	79.15%	72.45%	60.37%
MCR^2 TRAINING	**91.16%**	**89.70%**	**88.18%**	**86.66%**	**84.30%**



(a) $\Delta R\big(\boldsymbol{Z}(\theta), \mathbf{\Pi}, \epsilon\big)$ (b) $R(\boldsymbol{Z}(\theta), \epsilon)$ (c) $R^c(\boldsymbol{Z}(\theta), \epsilon \mid \mathbf{\Pi})$

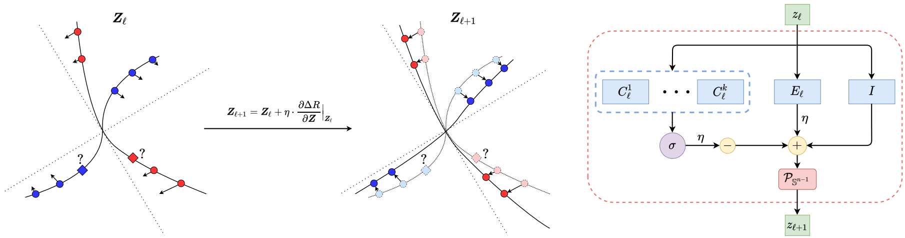
Figure: Evolution of $R, R^c, \Delta R$ of MCR^2 during training with corrupted labels.

Represent only what can be jointly compressed.

ReduNet: A White-box Deep Network from MCR^2

A white-box, forward-constructed, deep neural network from maximizing the rate reduction based on projected gradient flow:

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



ReduNet: A Whitebox Deep Network from Rate Reduction (JMLR'21): https://arxiv.org/abs/2105.10446

From One-sided to Bi-directional Representation

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

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

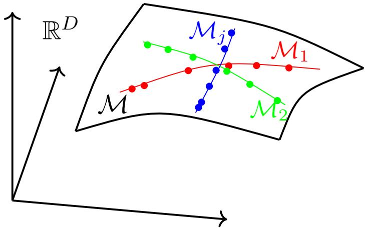
- Need to choose a proper feature dimension d.
- How good are the learned representation $\boldsymbol{Z}$?
- Anything missing, anything unexpected: $\dim(\boldsymbol{X}) = \dim(\boldsymbol{Z})$?
- Can we go from the feature $\boldsymbol{Z}$ back to the data $\boldsymbol{X}$?
- Is an LDR adequate to generate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

Can we find a bi-directional (auto-encoding) data representation:

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

Low-dim Representation for High-Dim Data

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



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

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

Problem Formulation

Desiderata for a good representation:

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

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

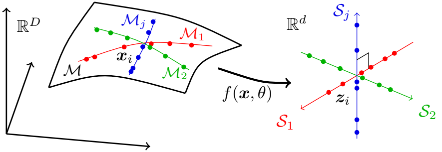
Caveats: we do not know $\dim(\mathcal{M})$ nor $d_j = \dim(\mathcal{M}_j)$. Often

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

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

- **So further wish the feature 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)

$$\boldsymbol{X} \subset \mathcal{S}^D \xrightarrow{\boldsymbol{V}^T} \boldsymbol{Z} \subset \mathcal{S}^d \xrightarrow{\boldsymbol{V}} \hat{\boldsymbol{X}} \subset \mathcal{S}^D. \tag{9}$$

Multiple linear subspaces: Generalized PCA (GPCA)[1]

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

One low-dim nonlinear submanifold: Nonlinear PCA[2]

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

The most general (likely the most important) case:

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

[1] Generalized principal component analysis, R. Vidal, Yi Ma, and S. Sastry, 2005.
[2] Nonlinear PCA using autoassociative neural networks, M. Krammer, 1991.

Principal Component Analysis (Auto Encoding)

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

$$\boldsymbol{X} \subset \mathcal{S}^D \xrightarrow{\boldsymbol{V}^T} \boldsymbol{Z} \subset \mathcal{S}^d \xrightarrow{\boldsymbol{V}} \hat{\boldsymbol{X}} \subset \mathcal{S}^D. \tag{13}$$

Solve the following optimization problem:

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

Principal Component Analysis (Auto Encoding)

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

$$\boldsymbol{X} \subset \mathcal{S}^D \xrightarrow{\boldsymbol{V}^T} \boldsymbol{Z} \subset \mathcal{S}^d \xrightarrow{\boldsymbol{V}} \hat{\boldsymbol{X}} \subset \mathcal{S}^D. \tag{13}$$

Solve the following optimization problem:

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

One low-dim nonlinear submanifold: Nonlinear PCA

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

Solve the following optimization problem:

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

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

Auto Encoding and its Difficulties

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

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

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

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

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

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

Instead optimize certain variational lower bounds (VAE):[3]

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

[3] 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):

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

A minimax game between generator and discriminator:

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

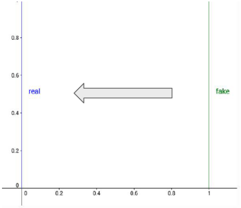
This is equivalent to minimize the *Jensen-Shannon divergence*:

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

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!)



Real and fake distribution when $\theta = 1$

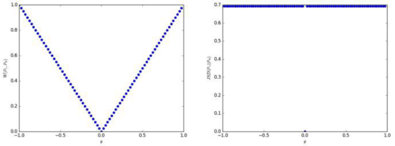


Figure 1: These plots show $\rho(\mathbb{P}_\theta, \mathbb{P}_0)$ as a function of θ when ρ is the EM distance (left plot) or the JS divergence (right plot). The EM plot is continuous and provides a usable gradient everywhere. The JS plot is not continuous and does not provide a usable gradient.

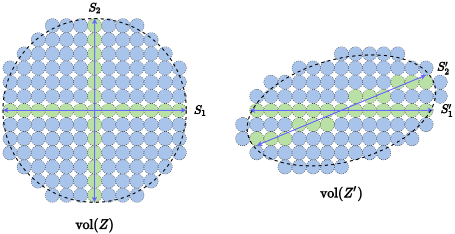
Replace $\mathcal{D}_{JS}$ with the *Earth-Mover* distance or *Wasserstein-1 distance*:

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

- 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 $\boldsymbol{Z}$, but what about $d(\boldsymbol{X}, \hat{\boldsymbol{X}})$?

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

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

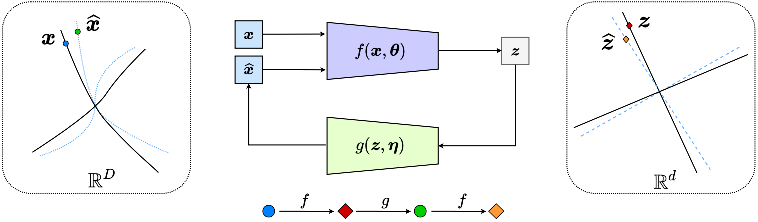
A New Closed-Loop Formulation

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

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

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

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



Measure Data Difference through Their Features

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

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

with the rate reduction measuring the error:

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

Measure Data Difference through Their Features

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

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

with the rate reduction measuring the error:

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

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

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

Measure Data Difference through Their Features

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

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

with the rate reduction measuring the error:

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

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

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

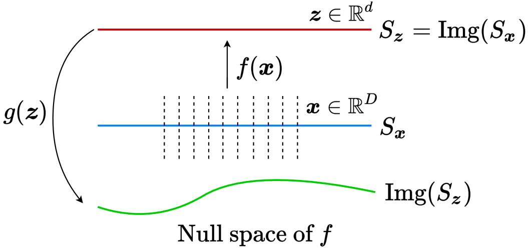
Encoder/sensor f amplifies any difference between $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$:

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

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 $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$.

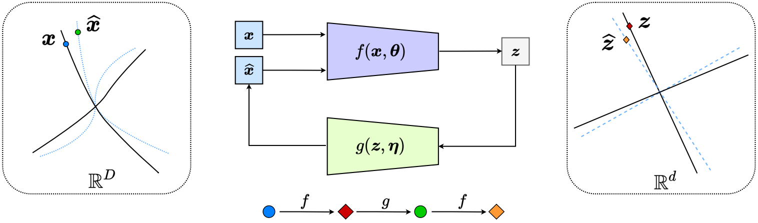
Reason: for a fixed encoder f, the decoder g can easily produce an ambiguous decoding such that the error between $\boldsymbol{Z}$ and $\hat{\boldsymbol{Z}}$ is zero!



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

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 $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ is given by a pursuit-evasion game between f as a "evader" and g as a "pursuer":

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

Overall Objective: Self-Consistency & Parsimony

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

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

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

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

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

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

Overall Objective: Self-Consistency & Parsimony

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

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

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

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

or equivalently

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

Characteristics of the Overall Objective

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

- **Simplicity:** all terms are uniformly rate reduction on features.
- **Excplicit:** distribution of learned features $\boldsymbol{Z}$ is an LDR.
- **A feedback loop** of encoding and decoding networks is all needed.
- **No** need or any direct explicit distance between $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$.
- **No** need to specify a prior or surrogate target distribution.
- **No** approximation by lower or upper bounds.
- **No** heuristics or regularizing terms.

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

Empirical Verification on Visual Data

Experimental Setup:

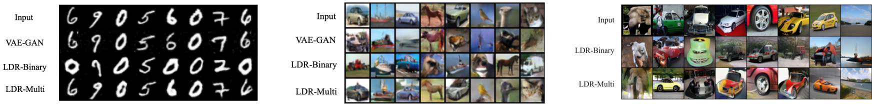
- **Datasets:** MNIST, CIFAR10, STL-10, CelebA faces, LSUN bedroom, ImageNet
- **Network architectures:** basic DCGAN & ResNet (not customized).
- **Feature space:** the same 128-dim regardless of data resolution or size
- **Quantization precision:** the same $\epsilon^2 = 0.5$.
- **Optimizer:** *Adam* 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 ↑	2.08	1.95	**2.21**	2.02	2.07
	FID ↓	24.78	20.15	33.65	**16.43**	16.47
CIFAR-10	IS ↑	7.32	7.23	7.11	**8.11**	7.13
	FID ↓	26.06	22.16	43.25	**19.63**	23.91

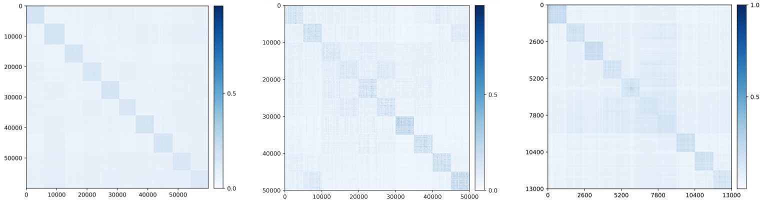
Table: Quantitative comparison on MNIST and CIFAR-10. Average Inception scores (IS) and FID scores. ↑ means higher is better. ↓ means lower is better.



(a) MNIST (b) CIFAR-10 (c) ImageNet

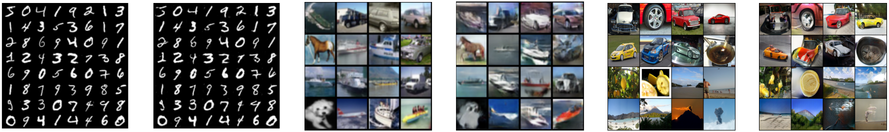
Figure: Qualitative comparison on MNIST, CIFAR-10 and ImageNet.

Empirical Verification on Visual Data



(a) MNIST (b) CIFAR10 (c) ImageNet

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



(a) MNIST $\boldsymbol{x}$ (b) MNIST $\hat{\boldsymbol{x}}$ (c) CIFAR10 $\boldsymbol{x}$ (d) CIFAR10 $\hat{\boldsymbol{x}}$ (e) ImageNet $\boldsymbol{x}$ (f) ImageNet $\hat{\boldsymbol{x}}$

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

Empirical Verification: Comparison on MNIST

(a) Original $\boldsymbol{X}$

(b) VAE-GAN $\hat{\boldsymbol{X}}$

(c) BiGAN $\hat{\boldsymbol{X}}$

(d) LDR-Binary $\hat{\boldsymbol{X}}$

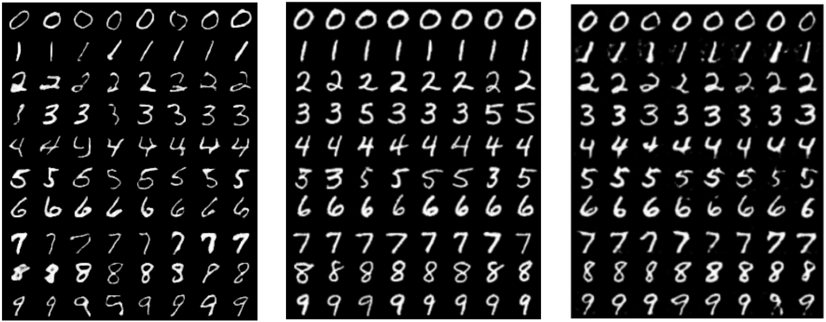
(e) LDR-Multi $\hat{\boldsymbol{X}}$

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

Empirical Verification: MNIST PCAs

The feature $\boldsymbol{z}$ in each of the k principal subspaces can be modeld as a degenerate Gaussian from the PCA $\boldsymbol{Z}_j = \boldsymbol{V}_j \boldsymbol{\Sigma}_j \boldsymbol{U}_j^T$:

$$\boldsymbol{z}_j \sim \bar{\boldsymbol{z}}_j + \sum_{l=1}^{r_j} n_l^j \sigma_j^l \boldsymbol{v}_j^l, \quad \text{where} \quad n_l^j \sim \mathcal{N}(0,1), \;\; j = 1, \ldots, k. \tag{31}$$



(a) ACGAN (b) InfoGAN (c) LDR-Multi

Empirical Verification: Interpolation between Samples

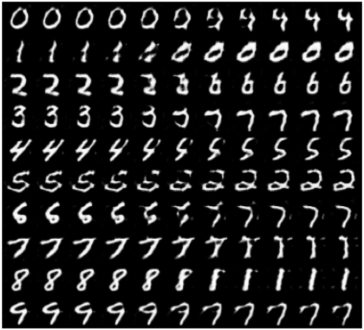
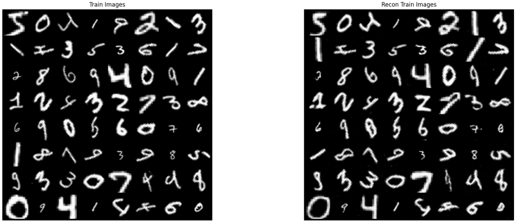
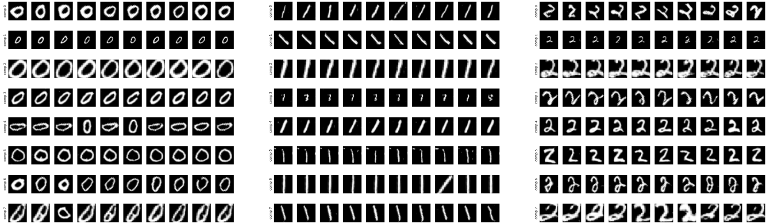


Figure: Images generated from interpolating between samples in different classes.

Empirical Verification: Transformed MNIST

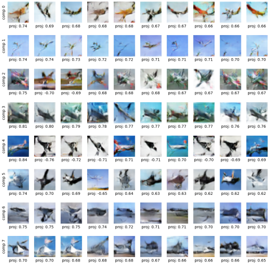
Original data $\boldsymbol{X}$ and their decoded version $\hat{\boldsymbol{X}}$ on transformed MNIST.

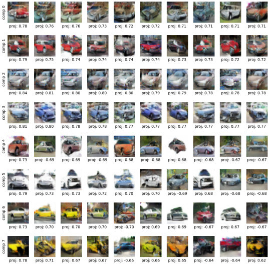


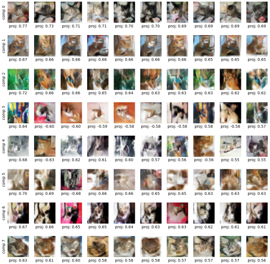


(c) Components of "0" (d) Components of "1" (e) Components of "2"

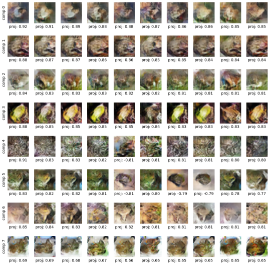
Empirical Verification: "Principal Images" of CIFAR10

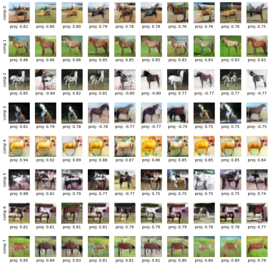


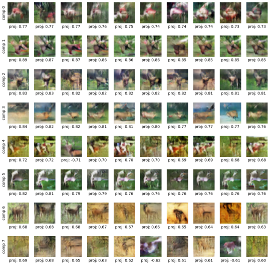




Empirical Verification: "Principal Images" of CIFAR10







Empirical Verification: “Principal Images” of CIFAR10

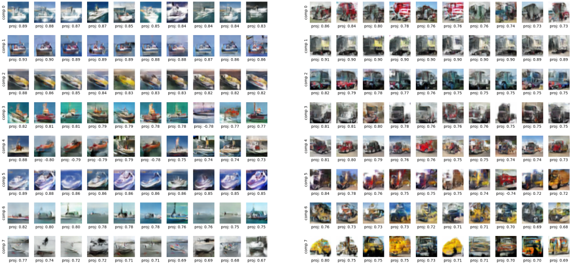


Figure: Reconstructed images $\hat{X}$ from features Z close to the principal components learned for each of the 10 classes of CIFAR-10.

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

Empirical Verification: Principal Components of CelebA

Visual attributes are disentangled as principal components.

(a) Hat

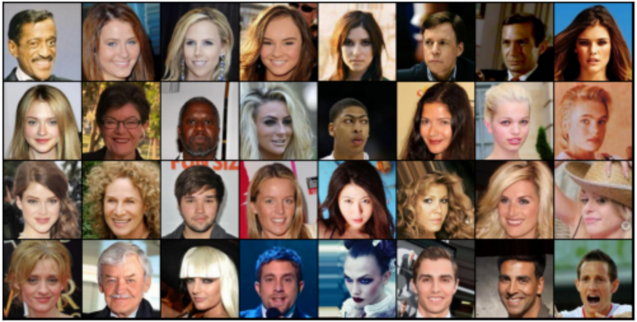
(b) Hair Color

(c) Glasses

Figure: Sampling along the 9-th, 19-th, and 23-th principal components of the learned features Z seems to manipulate the visual attributes for generated images, on the CelebA dataset.

Empirical Verification: CelebA Random Generation

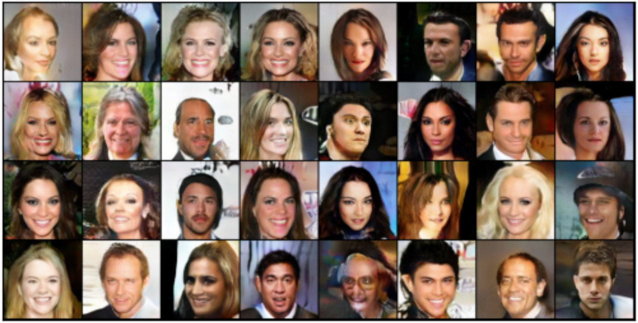
Empirical Verification: CelebA Input $\boldsymbol{X}$



(a) Original $\boldsymbol{X}$

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

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



(a) Decoded $\hat{\boldsymbol{X}}$

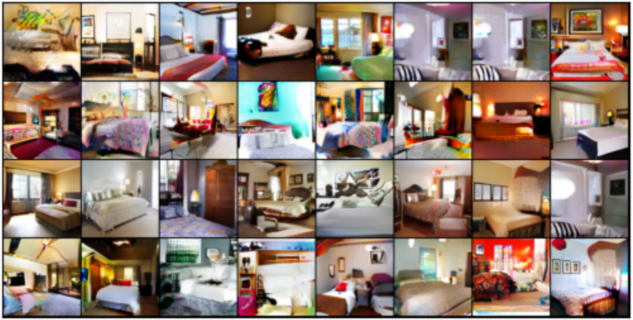
Figure: Visualizing the original $\boldsymbol{x}$ and corresponding decoded $\hat{\boldsymbol{x}}$ results on Celeb-A dataset. The LDR model is trained from LDR-Binary.

Empirical Verification: LSUN Bedroom Input $\boldsymbol{X}$

(a) Original $\boldsymbol{X}$

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

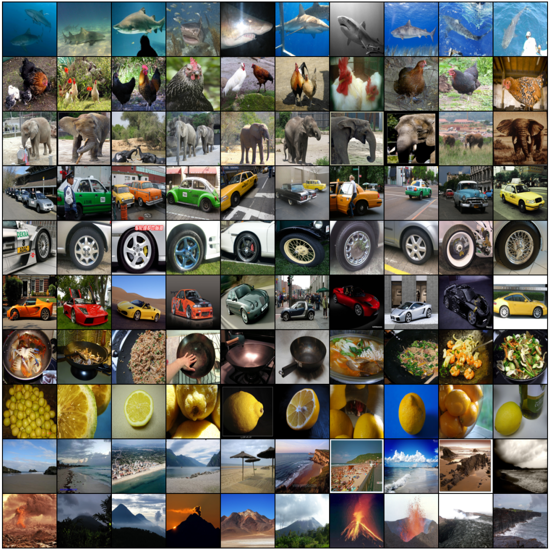
Empirical Verification: LSUN Bedroom Decoded $\hat{\boldsymbol{X}}$



(a) Decoded $\hat{\boldsymbol{X}}$

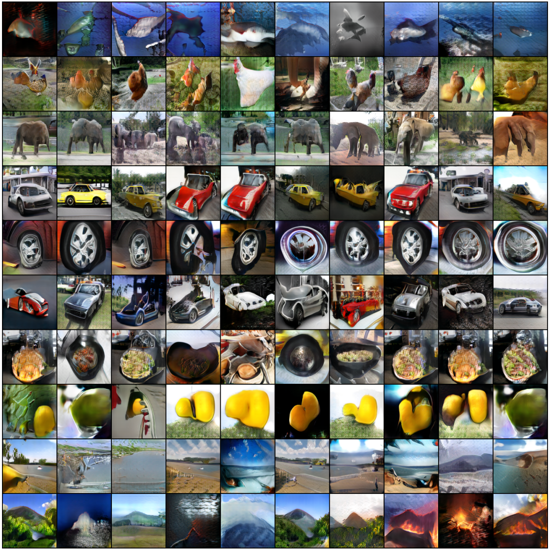
Figure: Visualizing the original $\boldsymbol{x}$ and corresponding decoded $\hat{\boldsymbol{x}}$ results on LSUN-bedroom dataset. The LDR model is trained from LDR-Binary.

Empirical Verification: ImageNet 10-Class Input $\boldsymbol{X}$



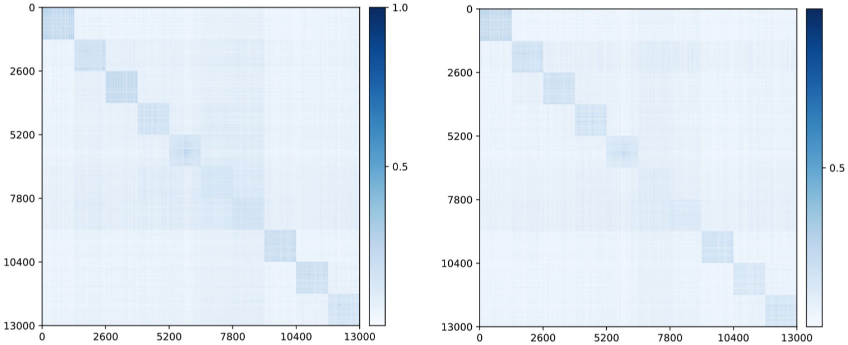
(a) Original $\boldsymbol{X}$

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



(b) Decoded $\hat{X}$

Empirical Verification: ImageNet Feature Similarity



(c) $|\boldsymbol{Z}^\top \boldsymbol{Z}|$ (d) $|\boldsymbol{Z}^\top \hat{\boldsymbol{Z}}|$

Figure: Visualizing feature alignment: (a) among features $|\boldsymbol{Z}^\top \boldsymbol{Z}|$, (b) between features and decoded features $|\boldsymbol{Z}^\top \hat{\boldsymbol{Z}}|$. 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↑	FID↓	IS↑	FID↓	IS↑	FID↓
GAN based methods						
DCGAN	6.6	-	7.8	-	-	-
SNGAN	7.4	29.3	**9.1**	40.1	-	48.73
CSGAN	8.1	19.6	-	-	-	-
LOGAN	**8.7**	**17.7**	-	-	-	-
VAE/GAN based methods						
VAE	3.8	115.8	-	-	-	-
VAE/GAN	7.4	39.8	-	-	-	-
NVAE	-	50.8	-	-	-	-
DC-VAE	**8.2**	**17.9**	8.1	41.9	-	-
LDR-Binary (ours)	**8.1**	**19.6**	8.4	**38.6**	7.74	**46.95**
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 #).

Empirical Verification: Other Ablation Studies

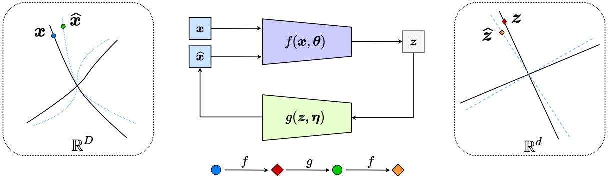
$$\min_{\eta} \max_{\theta} \Delta R\big(\boldsymbol{Z}(\theta)\big) + \Delta R\big(\hat{\boldsymbol{Z}}(\theta, \eta)\big) + \sum_{j=1}^{k} \Delta R\big(\boldsymbol{Z}_j(\theta), \hat{\boldsymbol{Z}}_j(\theta, \eta)\big).$$

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:** without the need for a distance or surrogate in the external data space.

Open Mathematical Problems

For the closed-loop minimax rate reduction program:

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

- **optimality:** characterization of the equilibrium points.
- **convergence** of the closed-loop control problem (infinite-dim).
- **deformable manifold learning** for the support of the distributions.
- **optimal density** of the distribution (*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?
- Better **feedback** for generative quality and discriminative property?
- **Whitebox** architectures for closed-loop transcription (ReduNet like)?
- Internal computational mechanisms for **memory** forming (Nature)?
- Closed-loop transcription to **other types of low-dim structures**? (dynamical, symbolical, logical, graphical...)

The principles of parsimony and feedback shall always rule!

References: Learning via Compression and Rate Reduction

1. **Closed-Loop** Data Transcription to an LDR via Minimaxing Rate Reduction
   https://arxiv.org/abs/2111.06636

2. **ReduNet:** A Whitebox Deep Network from Rate Reduction (JMLR'21):
   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

5. **Clustering** via Lossy Coding and Compression (TPAMI 2007):
   http://people.eecs.berkeley.edu/~yima/psfile/Ma-PAMI07.pdf

Parsimony and feedback are all you need to learn a compact and simple model for real-world data?

Thank you!

Questions, please?