From Information Theory to Machine Learning Algorithms: A Few Vignettes

Jongha (Jon) Ryu
UC San Diego

Ph.D. final defense

June 3, 2022

From information theory to machine learning algorithms

- Information theory studies how to communicate over or compress distributions

From information theory to machine learning algorithms

- Information theory studies how to communicate over or compress distributions
  - fundamental limits (achievability and converse);

From information theory to machine learning algorithms

- Information theory studies how to communicate over or compress distributions
  - fundamental limits (achievability and converse);
  - coding schemes;

From information theory to machine learning algorithms

- Information theory studies how to communicate over or compress distributions
  - fundamental limits (achievability and converse);
  - coding schemes;
  - information measures;

From information theory to machine learning algorithms

- Information theory studies how to communicate over or compress distributions
  - fundamental limits (achievability and converse);
  - coding schemes;
  - information measures;
  - mathematical tools; ...

From information theory to machine learning algorithms

- Information theory studies how to communicate over or compress distributions
  - fundamental limits (achievability and converse);
  - coding schemes;
  - information measures;
  - mathematical tools; ...

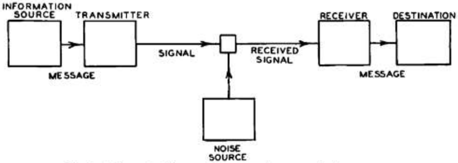


Fig. 1—Schematic diagram of a general communication system.

From information theory to machine learning algorithms

- Information theory studies how to communicate over or compress distributions
- Machine learning studies how to learn (about) distributions from their samples

From information theory to machine learning algorithms

- Information theory studies how to communicate over or compress distributions
- Machine learning studies how to learn (about) distributions from their samples

Q. How can we use tools and lessons from information theory
to develop machine learning algorithms?

From information theory to machine learning algorithms

- Information theory studies how to communicate over or compress distributions
- Machine learning studies how to learn (about) distributions from their samples

> **Q.** How can we use tools and lessons from information theory
> to develop machine learning algorithms?

- A few information-theoretic strategies to approach to a learning problem:

From information theory to machine learning algorithms

- Information theory studies how to communicate over or compress distributions
- Machine learning studies how to learn (about) distributions from their samples

> **Q.** How can we use tools and lessons from information theory
> to develop machine learning algorithms?

- A few information-theoretic strategies to approach to a learning problem:
  - abstract out the gist from it in the infinite-sample limit;

From information theory to machine learning algorithms

- Information theory studies how to communicate over or compress distributions
- Machine learning studies how to learn (about) distributions from their samples

Q. How can we use tools and lessons from information theory
to develop machine learning algorithms?

- A few information-theoretic strategies to approach to a learning problem:
  - abstract out the gist from it in the infinite-sample limit;
  - reduce it to a probability estimation problem and plug-in a "good" probability;

From information theory to machine learning algorithms

- Information theory studies how to communicate over or compress distributions
- Machine learning studies how to learn (about) distributions from their samples

Q. How can we use tools and lessons from information theory
to develop machine learning algorithms?

- A few information-theoretic strategies to approach to a learning problem:
  - abstract out the gist from it in the infinite-sample limit;
  - reduce it to a probability estimation problem and plug-in a "good" probability;
  - adapt and apply relevant ideas from information theory,
    e.g., Wyner's common information, context-tree weighting, mixture probability, ...

A few vignettes

1. Representation learning

2. Nonparametric methods for large-scale data

3. Assumption-free data processing

A few vignettes

1. Representation learning
   - learning a generative model with succinct representation learning [Ryu+21];
   - a fast kernel embedding without matrix eigendecomposition [RHK21];
   - unifying and generalizing contrastive representation learning methods [in progress]
2. Nonparametric methods for large-scale data
3. Assumption-free data processing

A few vignettes

1. Representation learning
   - learning a generative model with succinct representation learning [Ryu+21];
   - a fast kernel embedding without matrix eigendecomposition [RHK21];
   - unifying and generalizing contrastive representation learning methods [in progress]
2. Nonparametric methods for large-scale data
   - optimal classification, regression [RK22], and density estimation [in progress] with 1-nearest neighbors;
   - consistent density-functional estimation with fixed-k-nearest neighbors [Ryu+22];
   - an online mode-estimation algorithm [in progress]
3. Assumption-free data processing

A few vignettes

1. Representation learning
   - learning a generative model with succinct representation learning [Ryu+21];
   - a fast kernel embedding without matrix eigendecomposition [RHK21];
   - unifying and generalizing contrastive representation learning methods [in progress]
2. Nonparametric methods for large-scale data
   - optimal classification, regression [RK22], and density estimation [in progress] with 1-nearest neighbors;
   - consistent density-functional estimation with fixed-k-nearest neighbors [Ryu+22];
   - an online mode-estimation algorithm [in progress]
3. Assumption-free data processing
   - efficient universal discrete denoising [RK18];
   - parameter-free online learning with side information via universal gambling [RBK22];
   - universal portfolio with continuous side information [BRK22]
   - time-uniform concentration inequality via universal gambling [in progress]

A few vignettes

1. Representation learning
   - learning a generative model with succinct representation learning [Ryu+21];
   - a fast kernel embedding without matrix eigendecomposition [RHK21];
   - unifying and generalizing contrastive representation learning methods [in progress]
2. Nonparametric methods for large-scale data
   - optimal classification, regression [RK22], and density estimation [in progress] with 1-nearest neighbors;
   - consistent density-functional estimation with fixed-k-nearest neighbors [Ryu+22];
   - an online mode-estimation algorithm [in progress]
3. Assumption-free data processing
   - efficient universal discrete denoising [RK18];
   - parameter-free online learning with side information via universal gambling [RBK22];
   - universal portfolio with continuous side information [BRK22];
   - time-uniform concentration inequality via universal gambling [in progress]

A few vignettes

1. Representation learning
   - **learning a generative model with succinct representation learning** [Ryu+21];
   - a fast kernel embedding without matrix eigendecomposition [RHK21];
   - unifying and generalizing contrastive representation learning methods [in progress]
2. Nonparametric methods for large-scale data
   - **optimal classification, regression** [RK22], and density estimation [in progress] **with 1-nearest neighbors**;
   - consistent density-functional estimation with fixed-k-nearest neighbors [Ryu+22];
   - an online mode-estimation algorithm [in progress]
3. Assumption-free data processing
   - efficient universal discrete denoising [RK18];
   - parameter-free online learning with side information via universal gambling [RBK22];
   - universal portfolio with continuous side information [BRK22];
   - time-uniform concentration inequality via universal gambling [in progress]

Part I

From Wyner's Common Information to Learning with Succinct Common Representation

Problem setting

- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations

Problem setting

- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations
- **Examples**: $\{($story$_i$, illustration$_i)\}$,

Alice was beginning to get very tired of sitting ...when suddenly a White Rabbit with pink eyes ran close by her ...see it pop down a large rabbit-hole under the hedge.

Problem setting

- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations
- **Examples**: $\{($story$_i$, illustration$_i)\}$, $\{($image$_i$, caption$_i)\}$, ...

the bird has a white body, black wings, and webbed orange feet

a blue bird with gray primaries and secondaries and white breast and throat

Problem setting

- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations
- **Examples**: $\{($story$_i$, illustration$_i)\}$, $\{($image$_i$, caption$_i)\}$, ...
- **Goal**: learn the data distribution and sample from it (a.k.a. generation)

Problem setting

- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations
- **Examples**: $\{($story$_i$, illustration$_i)\}$, $\{($image$_i$, caption$_i)\}$, ...
- **Goal**: learn the data distribution and sample from it (a.k.a. generation)
  - Joint generation: Learn $q(\mathbf{x}, \mathbf{y})$ and generate $(\mathbf{X}, \mathbf{Y})$

Problem setting

- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations
- **Examples**: $\{($story$_i$, illustration$_i)\}$, $\{($image$_i$, caption$_i)\}$, ...
- **Goal**: learn the data distribution and sample from it (a.k.a. generation)
  - Joint generation: Learn $q(\mathbf{x}, \mathbf{y})$ and generate $(\mathbf{X}, \mathbf{Y})$
  - Conditional generation: Learn $q(\mathbf{y}|\mathbf{x})$ and generate $\mathbf{Y}$ given $\mathbf{x} \sim q(\mathbf{x})$

Problem setting

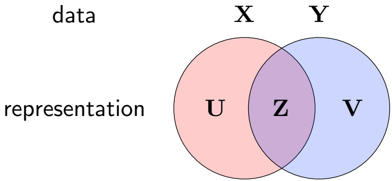
- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations
- **Examples**: $\{($story$_i$, illustration$_i)\}$, $\{($image$_i$, caption$_i)\}$, ...
- **Goal**: learn the data distribution and sample from it (a.k.a. generation)
  - Joint generation: Learn $q(\mathbf{x}, \mathbf{y})$ and generate $(\mathbf{X}, \mathbf{Y})$
  - Conditional generation: Learn $q(\mathbf{y}|\mathbf{x})$ and generate $\mathbf{Y}$ given $\mathbf{x} \sim q(\mathbf{x})$
  - Cross-domain retrieval: Given a query $\mathbf{x}$, retrieve relevant $\mathbf{y}$'s from a pool $\{\mathbf{y}_i\}_{i=1}^n$

Problem setting

- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations
- **Examples**: $\{($story$_i$, illustration$_i)\}$, $\{($image$_i$, caption$_i)\}$, ...
- **Goal**: learn the data distribution and sample from it (a.k.a. generation)
- Fit a generative model with structured latent representations $(\mathbf{Z}, \mathbf{U}, \mathbf{V})$

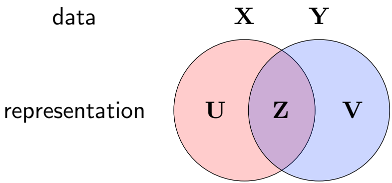
Problem setting

- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations
- **Examples**: $\{($story$_i$, illustration$_i)\}$, $\{($image$_i$, caption$_i)\}$, ...
- **Goal**: learn the data distribution and sample from it (a.k.a. generation)
- Fit a generative model with structured latent representations $(\mathbf{Z}, \mathbf{U}, \mathbf{V})$
  - Disentangle commonality $\mathbf{Z}$ from private properties $\mathbf{U}$ and $\mathbf{V}$



Problem setting

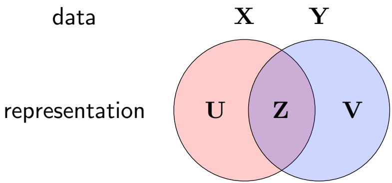
- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations
- **Examples**: $\{($story$_i$, illustration$_i)\}$, $\{($image$_i$, caption$_i)\}$, ...
- **Goal**: learn the data distribution and sample from it (a.k.a. generation)
- Fit a generative model with structured latent representations $(\mathbf{Z}, \mathbf{U}, \mathbf{V})$
  - Disentangle commonality $\mathbf{Z}$ from private properties $\mathbf{U}$ and $\mathbf{V}$
  - a.k.a. cross-domain disentanglement problem



Problem setting

- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations
- **Examples**: $\{($story$_i$, illustration$_i)\}$, $\{($image$_i$, caption$_i)\}$, ...
- **Goal**: learn the data distribution and sample from it (a.k.a. generation)
- Fit a generative model with structured latent representations $(\mathbf{Z}, \mathbf{U}, \mathbf{V})$

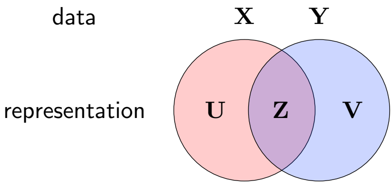
Q. Under which criterion should we disentangle?



Problem setting

- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations
- **Examples**: $\{($story$_i$, illustration$_i)\}$, $\{($image$_i$, caption$_i)\}$, ...
- **Goal**: learn the data distribution and sample from it (a.k.a. generation)
- Fit a generative model with structured latent representations $(\mathbf{Z}, \mathbf{U}, \mathbf{V})$

Q. What is an optimal common representation?

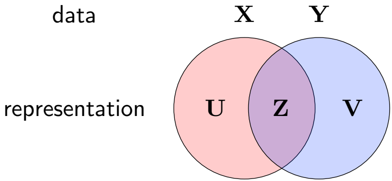


Problem setting

- **Data**: $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$ i.i.d. $\sim q(\mathbf{x}, \mathbf{y})$; high. dim., many-to-many relations
- **Examples**: $\{($story$_i$, illustration$_i)\}$, $\{($image$_i$, caption$_i)\}$, ...
- **Goal**: learn the data distribution and sample from it (a.k.a. generation)
- Fit a generative model with structured latent representations $(\mathbf{Z}, \mathbf{U}, \mathbf{V})$

Q. What is an optimal common representation?

- **A**. Use information theory to learn disentangled representations!



Motivation

- Cooperative game between Alice and Bob

Motivation

- Cooperative game between Alice and Bob

Motivation

- Cooperative game between Alice and Bob

Motivation

- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo

$\mathbf{X}$

Child's photo

Motivation

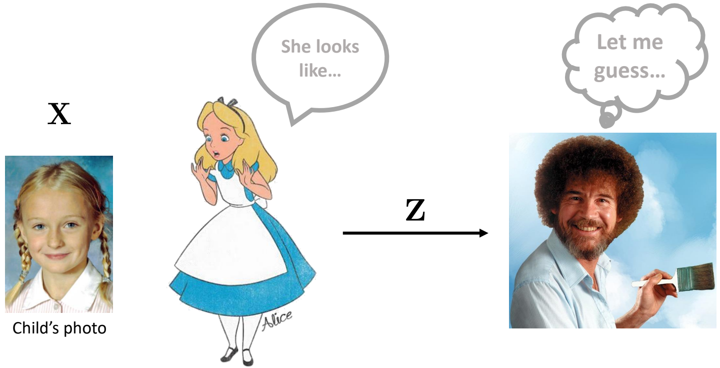
- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo

$\mathbf{X}$

Child's photo

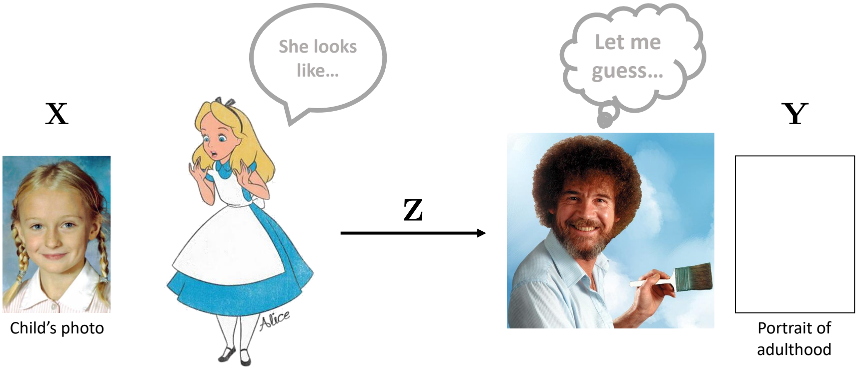
Motivation

- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo



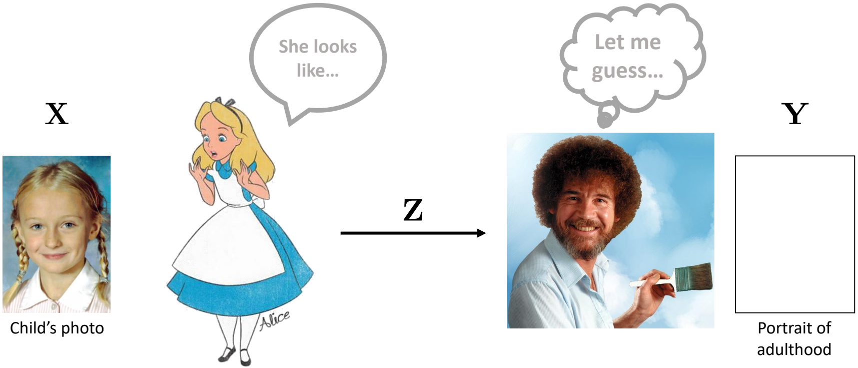
Motivation

- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo



Motivation

- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo
- What description dose Alice need to generate and send to help Bob?

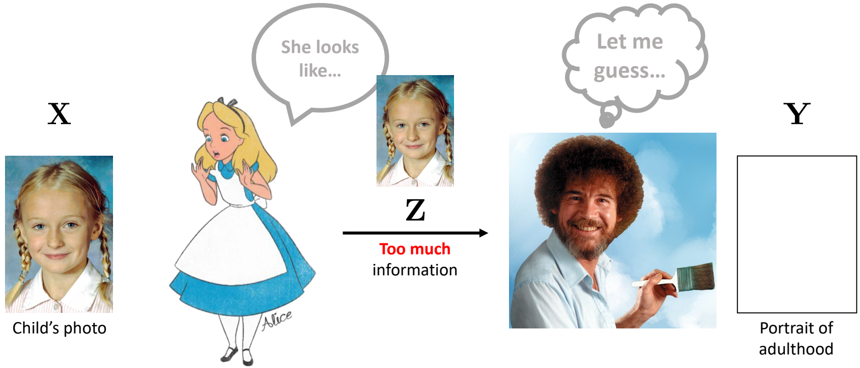


Motivation

- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo
- What description dose Alice need to generate and send to help Bob?

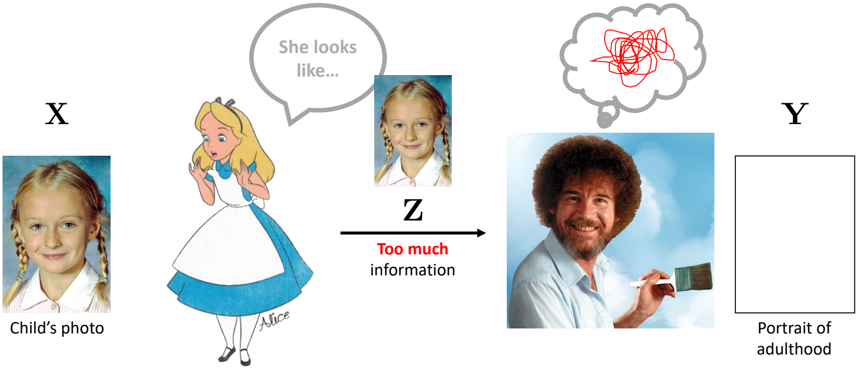
Motivation

- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo
- What description dose Alice need to generate and send to help Bob?

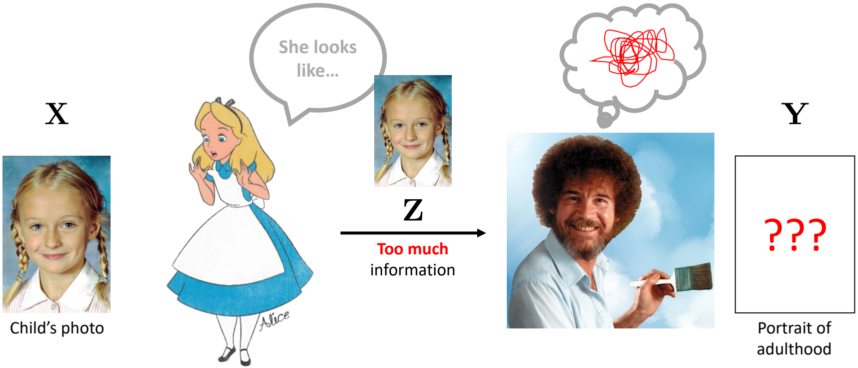


Motivation

- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo
- What description dose Alice need to generate and send to help Bob?

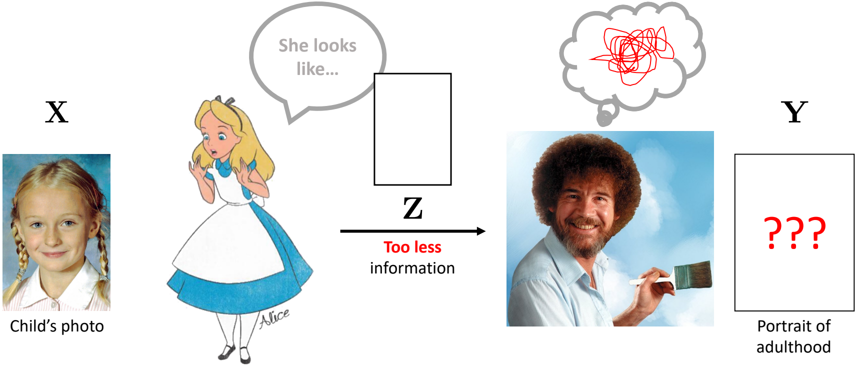


- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo
- What description dose Alice need to generate and send to help Bob?



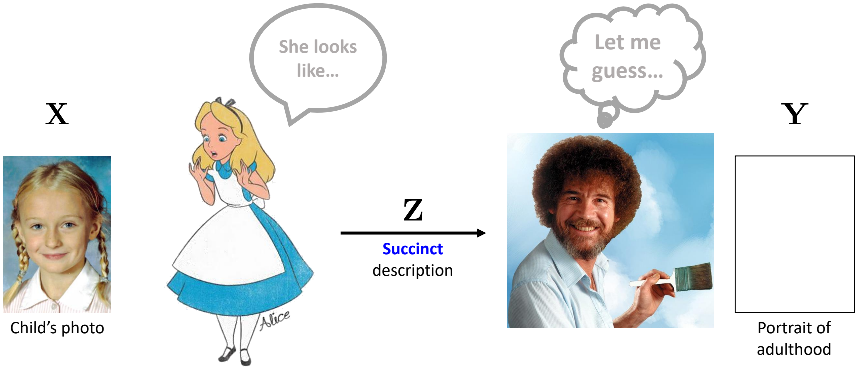
Motivation

- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo
- What description dose Alice need to generate and send to help Bob?



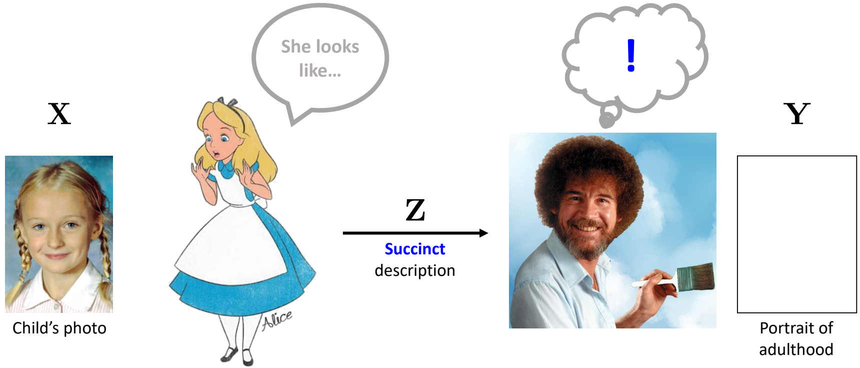
Motivation

- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo
- What description dose Alice need to generate and send to help Bob?



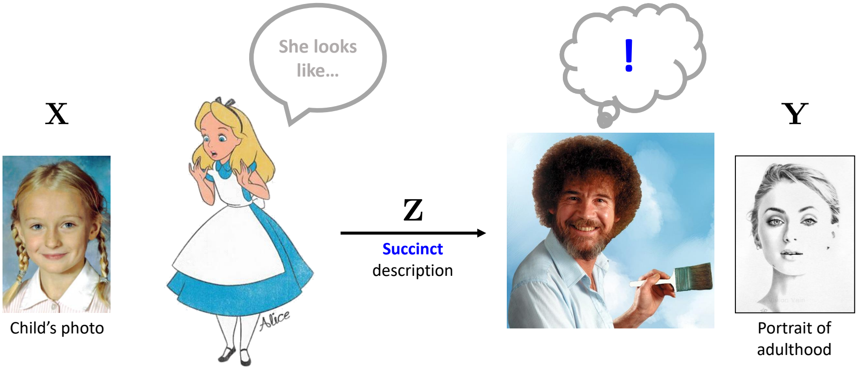
Motivation

- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo
- What description dose Alice need to generate and send to help Bob?



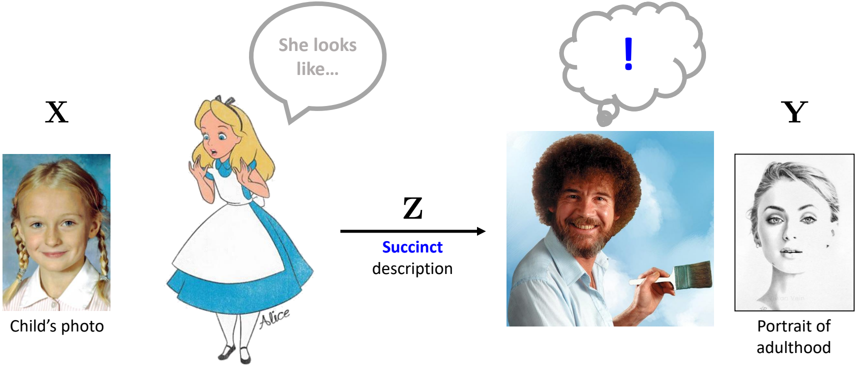
Motivation

- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo
- What description dose Alice need to generate and send to help Bob?



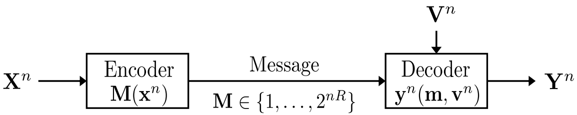
Motivation

- Cooperative game between Alice and Bob
- Alice and Bob wish to draw a nice portrait of adulthood from a child's photo
- What description dose Alice need to generate and send to help Bob?
- Alice can maximally help Bob by providing the most "succinct" description!



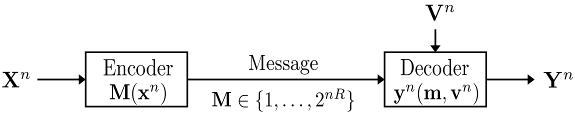
Channel synthesis (Cuff 2013)

- **Problem**: simulate a channel $q(\mathbf{y}|\mathbf{x})$ by communicating nR bits



Channel synthesis (Cuff 2013)

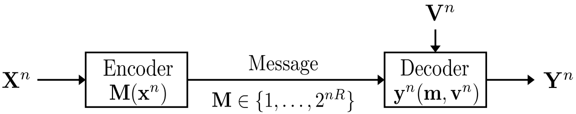
- **Problem**: simulate a channel $q(\mathbf{y}|\mathbf{x})$ by communicating nR bits



- **Question**: What is the minimum rate R^*?

Channel synthesis (Cuff 2013)

- **Problem**: simulate a channel $q(\mathbf{y}|\mathbf{x})$ by communicating nR bits

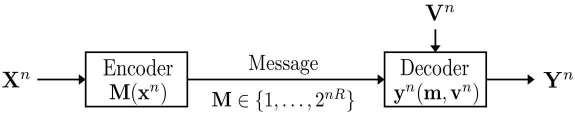


- **Question**: What is the minimum rate R^*?

- **Answer**: Wyner's common information $R^* = J(\mathbf{X}; \mathbf{Y})$

Channel synthesis (Cuff 2013)

- **Problem**: simulate a channel $q(\mathbf{y}|\mathbf{x})$ by communicating nR bits



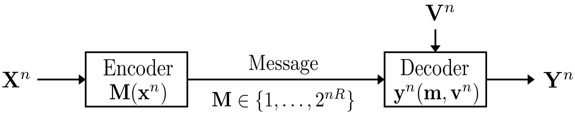
- **Question**: What is the minimum rate R^*?

- **Answer**: Wyner's common information $R^* = J(\mathbf{X}; \mathbf{Y})$

minimize	$I(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$
subject to	$\mathbf{X} - \mathbf{Z} - \mathbf{Y}$
variables	$q(\mathbf{z}\|\mathbf{x}, \mathbf{y})$

Channel synthesis (Cuff 2013)

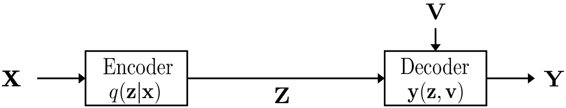
- **Problem**: simulate a channel $q(\mathbf{y}|\mathbf{x})$ by communicating nR bits



- **Question**: What is the minimum rate R^*?

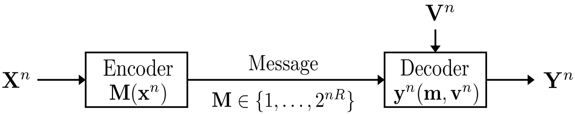
- **Answer**: Wyner's common information $R^* = J(\mathbf{X}; \mathbf{Y})$

- Single-letter characterization



Channel synthesis (Cuff 2013)

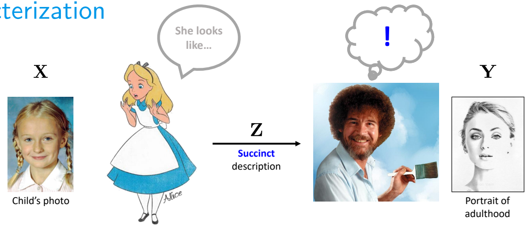
- **Problem**: simulate a channel $q(\mathbf{y}|\mathbf{x})$ by communicating nR bits



- **Question**: What is the minimum rate R^*?

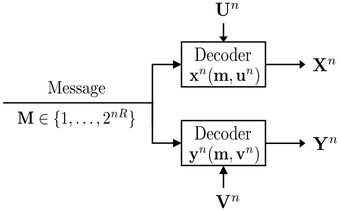
- **Answer**: Wyner's common information $R^* = J(\mathbf{X}; \mathbf{Y})$

- Single-letter characterization



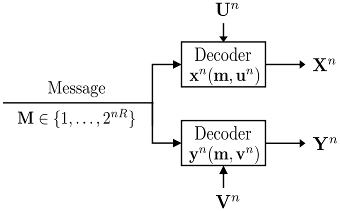
Distributed simulation (Wyner 1975)

- **Problem**: simulate a joint distribution $q(\mathbf{x}, \mathbf{y})$ from nR common bits



Distributed simulation (Wyner 1975)

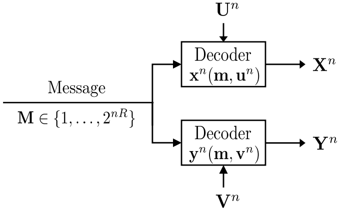
- **Problem**: simulate a joint distribution $q(\mathbf{x}, \mathbf{y})$ from nR common bits



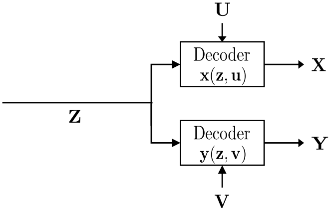
- **Question**: What is the minimum rate R^{**}?

Distributed simulation (Wyner 1975)

- **Problem**: simulate a joint distribution $q(\mathbf{x}, \mathbf{y})$ from nR common bits



- **Question**: What is the minimum rate R^{**}?
- **Answer**: $R^{**} = J(\mathbf{X}; \mathbf{Y})$



Learning distributions based on Wyner's common information

- Channel synthesis $\rightarrow$ conditional generation
- Distributed simulation $\rightarrow$ joint generation

Learning distributions based on Wyner's common information

- Channel synthesis $\to$ conditional generation
- Distributed simulation $\to$ joint generation

Definition

Given $q(\mathbf{x}, \mathbf{y})$, define Wyner's common representation as a solution of

$$\boxed{\begin{array}{ll} \text{minimize} & I(\mathbf{X}, \mathbf{Y}; \mathbf{Z}) \\ \text{subject to} & \mathbf{X} - \mathbf{Z} - \mathbf{Y} \\ \text{variables} & q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) \end{array}}$$

Learning distributions based on Wyner's common information

- Channel synthesis $\to$ conditional generation
- Distributed simulation $\to$ joint generation

Definition

Given $q(\mathbf{x}, \mathbf{y})$, define Wyner's common representation as a solution of

$$\boxed{\begin{array}{ll} \text{minimize} & I(\mathbf{X}, \mathbf{Y}; \mathbf{Z}) \\ \text{subject to} & \mathbf{X} - \mathbf{Z} - \mathbf{Y} \\ \text{variables} & q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) \end{array}}$$

- Call Wyner's optimization problem

Learning distributions based on Wyner's common information

- Channel synthesis $\rightarrow$ conditional generation
- Distributed simulation $\rightarrow$ joint generation

Definition

Given $q(\mathbf{x}, \mathbf{y})$, define Wyner's common representation as a solution of

minimize	$I(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$
subject to	$\mathbf{X} - \mathbf{Z} - \mathbf{Y}$
variables	$q_\phi(\mathbf{z}\|\mathbf{x}, \mathbf{y})$

- Call Wyner's optimization problem
- $I(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ quantifies the complexity of the representation $\mathbf{Z}$

Learning distributions based on Wyner's common information

- Channel synthesis $\rightarrow$ conditional generation
- Distributed simulation $\rightarrow$ joint generation

Definition

Given $q(\mathbf{x}, \mathbf{y})$, define Wyner's common representation as a solution of

$$\boxed{\begin{array}{ll} \text{minimize} & I(\mathbf{X}, \mathbf{Y}; \mathbf{Z}) \\ \text{subject to} & \mathbf{X} - \mathbf{Z} - \mathbf{Y} \\ \text{variables} & q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) \end{array}}$$

- Call Wyner's optimization problem
- $I(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ quantifies the complexity of the representation $\mathbf{Z}$

- Given samples, let's learn a generative model with Wyner's common representation

Learning distributions based on Wyner's common information

- Channel synthesis $\to$ conditional generation
- Distributed simulation $\to$ joint generation

Definition

Given $q(\mathbf{x}, \mathbf{y})$, define Wyner's common representation as a solution of

$$\boxed{\begin{array}{ll} \text{minimize} & I(\mathbf{X}, \mathbf{Y}; \mathbf{Z}) \\ \text{subject to} & \mathbf{X} - \mathbf{Z} - \mathbf{Y} \\ \text{variables} & q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) \end{array}}$$

- Call Wyner's optimization problem
- $I(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ quantifies the complexity of the representation $\mathbf{Z}$

- Given samples, let's learn a generative model with Wyner's common representation
  - Consider the latent variable models induced by the single letter characterizations

Learning distributions based on Wyner's common information

- Channel synthesis $\to$ conditional generation
- Distributed simulation $\to$ joint generation

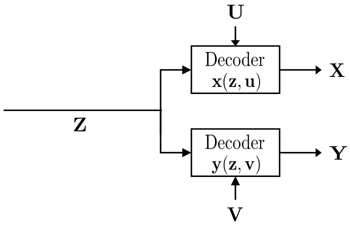
Definition

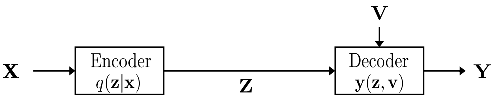
Given $q(\mathbf{x}, \mathbf{y})$, define Wyner's common representation as a solution of

$$\boxed{\begin{array}{ll} \text{minimize} & I(\mathbf{X}, \mathbf{Y}; \mathbf{Z}) \\ \text{subject to} & \mathbf{X} - \mathbf{Z} - \mathbf{Y} \\ \text{variables} & q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) \end{array}}$$

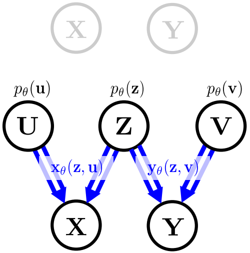
  - Call Wyner's optimization problem
  - $I(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ quantifies the complexity of the representation $\mathbf{Z}$
- Given samples, let's learn a generative model with Wyner's common representation
  - Consider the latent variable models induced by the single letter characterizations
  - Fit the generative models to data based on Wyner's optimization problem

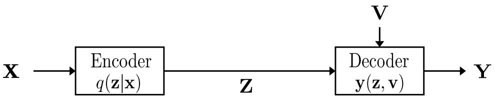
Probabilistic model



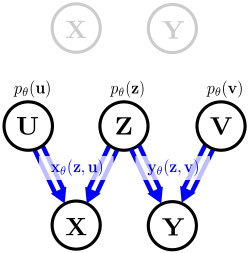


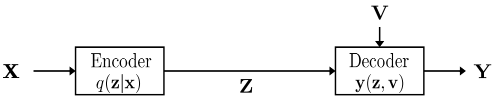
Probabilistic model





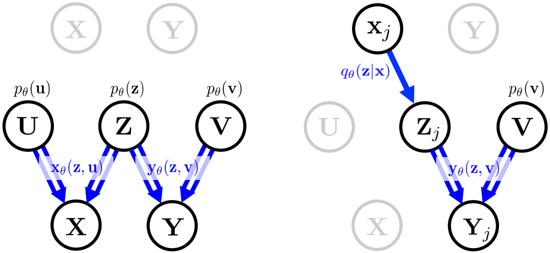
Probabilistic model





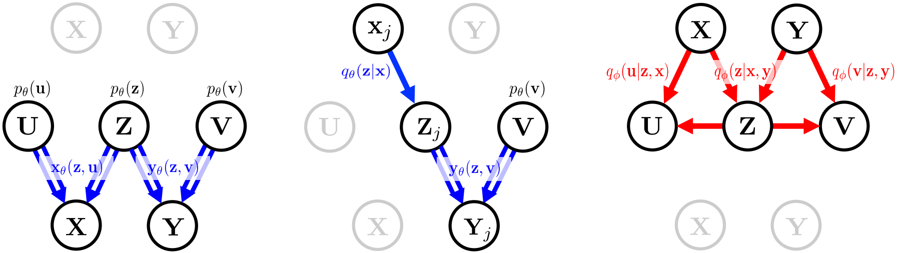
- Decoders: $\mathbf{x}_\theta(\mathbf{z},\mathbf{u}), \mathbf{y}_\theta(\mathbf{z},\mathbf{v})$
- Priors (source of randomness): common $p_\theta(\mathbf{z})$, local $p_\theta(\mathbf{u}), p_\theta(\mathbf{v})$

Probabilistic model



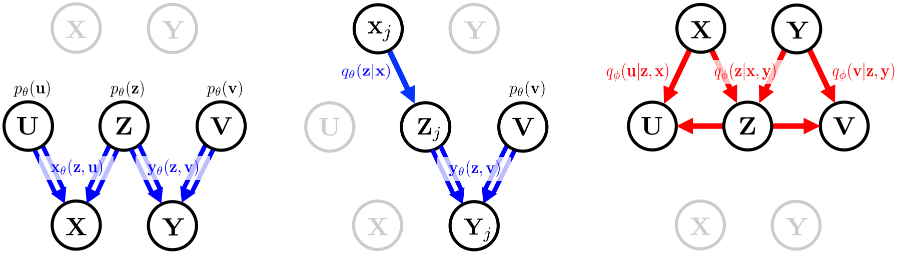
- Decoders: $\mathbf{x}_\theta(\mathbf{z},\mathbf{u}), \mathbf{y}_\theta(\mathbf{z},\mathbf{v})$
- Priors (source of randomness): common $p_\theta(\mathbf{z})$, local $p_\theta(\mathbf{u}), p_\theta(\mathbf{v})$
- Model (marginal) encoders: $q_\theta(\mathbf{z}|\mathbf{x})$, $q_\theta(\mathbf{z}|\mathbf{y})$

Probabilistic model



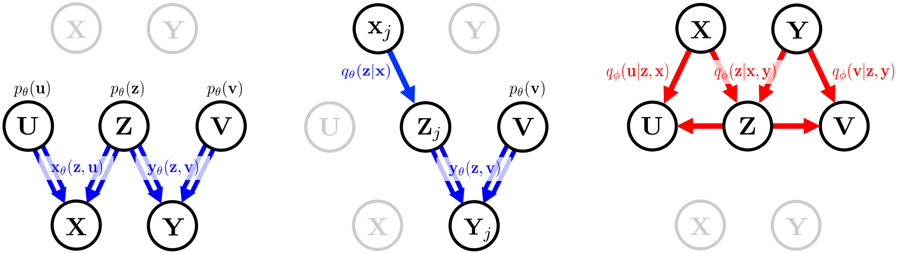
- Decoders: $\mathbf{x}_\theta(\mathbf{z},\mathbf{u}), \mathbf{y}_\theta(\mathbf{z},\mathbf{v})$
- Priors (source of randomness): common $p_\theta(\mathbf{z})$, local $p_\theta(\mathbf{u}), p_\theta(\mathbf{v})$
- Model (marginal) encoders: $q_\theta(\mathbf{z}|\mathbf{x})$, $q_\theta(\mathbf{z}|\mathbf{y})$
- Variational encoders: joint $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$, local $q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x})$, $q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y})$

Probabilistic model



- Decoders: $\mathbf{x}_\theta(\mathbf{z},\mathbf{u}), \mathbf{y}_\theta(\mathbf{z},\mathbf{v})$
- Priors (source of randomness): common $p_\theta(\mathbf{z})$, local $p_\theta(\mathbf{u}), p_\theta(\mathbf{v})$
- Model (marginal) encoders: $q_\theta(\mathbf{z}|\mathbf{x})$, $q_\theta(\mathbf{z}|\mathbf{y})$
- Variational encoders: joint $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$, local $q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x})$, $q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y})$
- Call these components in entirety the variational Wyner model

Probabilistic model

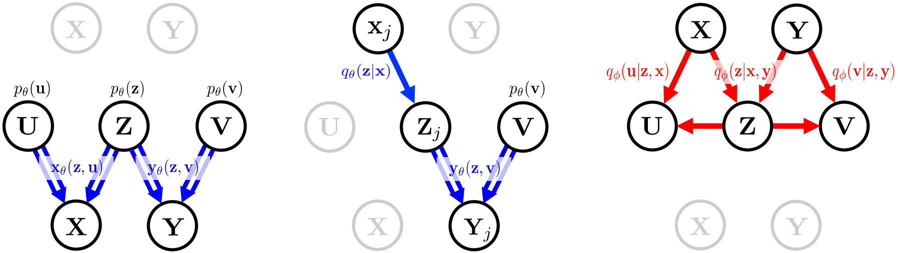


- Decoders: $\mathbf{x}_\theta(\mathbf{z},\mathbf{u}), \mathbf{y}_\theta(\mathbf{z},\mathbf{v})$
- Priors (source of randomness): common $p_\theta(\mathbf{z})$, local $p_\theta(\mathbf{u}), p_\theta(\mathbf{v})$
- Model (marginal) encoders: $q_\theta(\mathbf{z}|\mathbf{x})$, $q_\theta(\mathbf{z}|\mathbf{y})$
- Variational encoders: joint $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$, local $q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x})$, $q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y})$
- Call these components in entirety the variational Wyner model

} model θ

} variational ϕ

Probabilistic model



- Decoders: $\mathbf{x}_\theta(\mathbf{z},\mathbf{u}), \mathbf{y}_\theta(\mathbf{z},\mathbf{v})$
- Priors (source of randomness): common $p_\theta(\mathbf{z})$, local $p_\theta(\mathbf{u}), p_\theta(\mathbf{v})$

} decoders p

- Model (marginal) encoders: $q_\theta(\mathbf{z}|\mathbf{x})$, $q_\theta(\mathbf{z}|\mathbf{y})$
- Variational encoders: joint $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$, local $q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x})$, $q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y})$

} encoders q

- Call these components in entirety the variational Wyner model

Training objectives

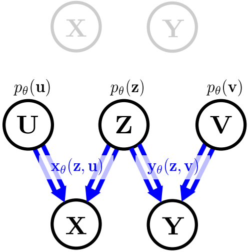
- The variational Wyner model induces four distributions:

joint
cond. $(x \to y)$
cond. $(y \to x)$
variational

Training objectives

- The variational Wyner model induces four distributions:

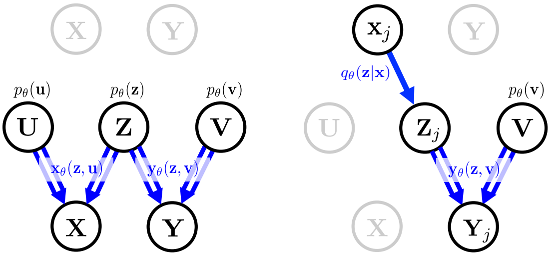
joint	$p_{\to\mathsf{xy}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \triangleq p_\theta(\mathbf{z})p_\theta(\mathbf{u})p_\theta(\mathbf{v})\delta(\mathbf{x}-\mathbf{x}_\theta(\mathbf{z},\mathbf{u}))\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. (x → y)	
cond. (y → x)	
variational	



Training objectives

- The variational Wyner model induces four distributions:

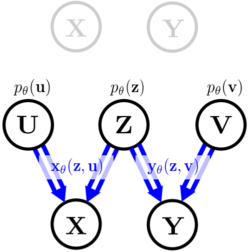
joint	$p_{\to \mathsf{xy}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \triangleq p_\theta(\mathbf{z})p_\theta(\mathbf{u})p_\theta(\mathbf{v})\delta(\mathbf{x}-\mathbf{x}_\theta(\mathbf{z},\mathbf{u}))\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. (x $\to$ y)	$p_{\mathsf{x}\to\mathsf{y}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{v}) \triangleq q(\mathbf{x})q_\theta(\mathbf{z}\vert\mathbf{x})p_\theta(\mathbf{v})\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. (y $\to$ x)	
variational	

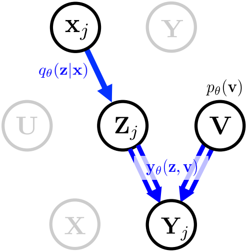


Training objectives

- The variational Wyner model induces four distributions:

joint	$p_{\rightarrow \mathsf{xy}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \triangleq p_\theta(\mathbf{z})p_\theta(\mathbf{u})p_\theta(\mathbf{v})\delta(\mathbf{x}-\mathbf{x}_\theta(\mathbf{z},\mathbf{u}))\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. $(\mathsf{x}\rightarrow\mathsf{y})$	$p_{\mathsf{x}\rightarrow\mathsf{y}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{v}) \triangleq q(\mathbf{x})q_\theta(\mathbf{z}\|\mathbf{x})p_\theta(\mathbf{v})\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. $(\mathsf{y}\rightarrow\mathsf{x})$	$p_{\mathsf{y}\rightarrow\mathsf{x}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u}) \triangleq q(\mathbf{y})q_\theta(\mathbf{z}\|\mathbf{y})p_\theta(\mathbf{u})\delta(\mathbf{x}-\mathbf{x}_\theta(\mathbf{z},\mathbf{u}))$
variational	

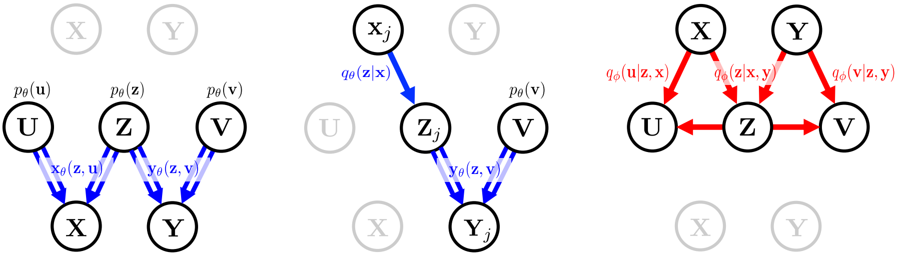




Training objectives

- The variational Wyner model induces four distributions:

joint	$p_{\to \mathsf{xy}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \triangleq p_\theta(\mathbf{z})p_\theta(\mathbf{u})p_\theta(\mathbf{v})\delta(\mathbf{x}-\mathbf{x}_\theta(\mathbf{z},\mathbf{u}))\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. $(\mathsf{x}\to\mathsf{y})$	$p_{\mathsf{x}\to\mathsf{y}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{v}) \triangleq q(\mathbf{x})q_\theta(\mathbf{z}\|\mathbf{x})p_\theta(\mathbf{v})\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. $(\mathsf{y}\to\mathsf{x})$	$p_{\mathsf{y}\to\mathsf{x}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u}) \triangleq q(\mathbf{y})q_\theta(\mathbf{z}\|\mathbf{y})p_\theta(\mathbf{u})\delta(\mathbf{x}-\mathbf{x}_\theta(\mathbf{z},\mathbf{u}))$
variational	$q_{\mathsf{xy}\to}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \triangleq q(\mathbf{x},\mathbf{y})q_\phi(\mathbf{z}\|\mathbf{x},\mathbf{y})q_\phi(\mathbf{u}\|\mathbf{z},\mathbf{x})q_\phi(\mathbf{v}\|\mathbf{z},\mathbf{y})$



Training objectives

- The variational Wyner model induces four distributions:

joint	$p_{\to\mathsf{xy}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \triangleq p_\theta(\mathbf{z})p_\theta(\mathbf{u})p_\theta(\mathbf{v})\delta(\mathbf{x}-\mathbf{x}_\theta(\mathbf{z},\mathbf{u}))\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. $(\mathsf{x}\to\mathsf{y})$	$p_{\mathsf{x}\to\mathsf{y}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{v}) \triangleq q(\mathbf{x})q_\theta(\mathbf{z}\vert\mathbf{x})p_\theta(\mathbf{v})\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. $(\mathsf{y}\to\mathsf{x})$	$p_{\mathsf{y}\to\mathsf{x}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u}) \triangleq q(\mathbf{y})q_\theta(\mathbf{z}\vert\mathbf{y})p_\theta(\mathbf{u})\delta(\mathbf{x}-\mathbf{x}_\theta(\mathbf{z},\mathbf{u}))$
variational	$q_{\mathsf{xy}\to}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \triangleq q(\mathbf{x},\mathbf{y})q_\phi(\mathbf{z}\vert\mathbf{x},\mathbf{y})q_\phi(\mathbf{u}\vert\mathbf{z},\mathbf{x})q_\phi(\mathbf{v}\vert\mathbf{z},\mathbf{y})$

- Recall Wyner's optimization problem:

$$\boxed{\begin{array}{ll} \text{minimize} & I(\mathbf{X},\mathbf{Y};\mathbf{Z}) \\ \text{subject to} & \mathbf{X}-\mathbf{Z}-\mathbf{Y} \\ \text{variables} & q_\phi(\mathbf{z}\vert\mathbf{x},\mathbf{y}) \end{array}}$$

Training objectives

- The variational Wyner model induces four distributions:

joint	$p_{\to \mathsf{xy}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \triangleq p_\theta(\mathbf{z})p_\theta(\mathbf{u})p_\theta(\mathbf{v})\delta(\mathbf{x}-\mathbf{x}_\theta(\mathbf{z},\mathbf{u}))\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. ($\mathsf{x}\to\mathsf{y}$)	$p_{\mathsf{x}\to\mathsf{y}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{v}) \triangleq q(\mathbf{x})q_\theta(\mathbf{z}\|\mathbf{x})p_\theta(\mathbf{v})\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. ($\mathsf{y}\to\mathsf{x}$)	$p_{\mathsf{y}\to\mathsf{x}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u}) \triangleq q(\mathbf{y})q_\theta(\mathbf{z}\|\mathbf{y})p_\theta(\mathbf{u})\delta(\mathbf{x}-\mathbf{x}_\theta(\mathbf{z},\mathbf{u}))$
variational	$q_{\mathsf{xy}\to}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \triangleq q(\mathbf{x},\mathbf{y})q_\phi(\mathbf{z}\|\mathbf{x},\mathbf{y})q_\phi(\mathbf{u}\|\mathbf{z},\mathbf{x})q_\phi(\mathbf{v}\|\mathbf{z},\mathbf{y})$

- Recall Wyner's optimization problem:

minimize	$I(\mathbf{X},\mathbf{Y};\mathbf{Z})$
subject to	$\mathbf{X}-\mathbf{Z}-\mathbf{Y}$
variables	$q_\phi(\mathbf{z}\|\mathbf{x},\mathbf{y})$

- For each model $p_{\mathsf{model}} \in \{p_{\to\mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$, we can relax the problem as

$$\text{minimize} \quad D(p_{\mathsf{model}}, q_{\mathsf{xy}\to}) + \lambda^{\mathsf{CI}}_{\mathsf{model}} I_{\mathsf{model}}(\mathbf{X},\mathbf{Y};\mathbf{Z})$$

Training objectives

- The variational Wyner model induces four distributions:

joint	$p_{\to\mathsf{xy}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \triangleq p_\theta(\mathbf{z})p_\theta(\mathbf{u})p_\theta(\mathbf{v})\delta(\mathbf{x}-\mathbf{x}_\theta(\mathbf{z},\mathbf{u}))\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. (x $\to$ y)	$p_{\mathsf{x}\to\mathsf{y}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{v}) \triangleq q(\mathbf{x})q_\theta(\mathbf{z}\vert\mathbf{x})p_\theta(\mathbf{v})\delta(\mathbf{y}-\mathbf{y}_\theta(\mathbf{z},\mathbf{v}))$
cond. (y $\to$ x)	$p_{\mathsf{y}\to\mathsf{x}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u}) \triangleq q(\mathbf{y})q_\theta(\mathbf{z}\vert\mathbf{y})p_\theta(\mathbf{u})\delta(\mathbf{x}-\mathbf{x}_\theta(\mathbf{z},\mathbf{u}))$
variational	$q_{\mathsf{xy}\to}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \triangleq q(\mathbf{x},\mathbf{y})q_\phi(\mathbf{z}\vert\mathbf{x},\mathbf{y})q_\phi(\mathbf{u}\vert\mathbf{z},\mathbf{x})q_\phi(\mathbf{v}\vert\mathbf{z},\mathbf{y})$

- Recall Wyner's optimization problem:

$$\boxed{\begin{array}{ll} \text{minimize} & I(\mathbf{X},\mathbf{Y};\mathbf{Z}) \\ \text{subject to} & \mathbf{X}-\mathbf{Z}-\mathbf{Y} \\ \text{variables} & q_\phi(\mathbf{z}\vert\mathbf{x},\mathbf{y}) \end{array}}$$

- For each model $p_{\mathsf{model}} \in \{p_{\to\mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$, we can relax the problem as

$$\boxed{\text{minimize} \quad D(p_{\mathsf{model}}, q_{\mathsf{xy}\to}) + \lambda^{\mathsf{CI}}_{\mathsf{model}} I_{\mathsf{model}}(\mathbf{X},\mathbf{Y};\mathbf{Z})}$$

- Distribution matching with CI regularization

Training method

- For each model $p_{\text{model}} \in \{p_{\to\text{xy}}, p_{\text{x}\to\text{y}}, p_{\text{y}\to\text{x}}\}$,

$$\boxed{\text{minimize} \quad D(p_{\text{model}}, q_{\text{xy}\to}) + \lambda^{\text{CI}}_{\text{model}} I_{\text{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})}$$

- Distribution matching with CI regularization

Training method

- For each model $p_{\mathsf{model}} \in \{p_{\to\mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$,

$$\boxed{\text{minimize} \quad D(p_{\mathsf{model}}, q_{\mathsf{xy}\to}) + \lambda^{\mathsf{CI}}_{\mathsf{model}} I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})}$$

- Distribution matching with CI regularization
- Symmetric KL divergence $D_{\mathsf{sym}}(p, q) \triangleq D_{\mathsf{KL}}(p \parallel q) + D_{\mathsf{KL}}(q \parallel p)$

Training method

- For each model $p_{\text{model}} \in \{p_{\to \text{xy}}, p_{\text{x}\to\text{y}}, p_{\text{y}\to\text{x}}\}$,

$$\boxed{\text{minimize} \quad D(p_{\text{model}}, q_{\text{xy}\to}) + \lambda^{\text{CI}}_{\text{model}} I_{\text{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})}$$

- Distribution matching with CI regularization
- Symmetric KL divergence $D_{\text{sym}}(p, q) \triangleq D_{\text{KL}}(p \parallel q) + D_{\text{KL}}(q \parallel p)$
- Variational density-ratio estimation technique [Pu+17]

Training method

- For each model $p_{\mathsf{model}} \in \{p_{\to \mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$,

$$\boxed{\text{minimize} \quad D(p_{\mathsf{model}}, q_{\mathsf{xy}\to}) + \lambda^{\mathsf{CI}}_{\mathsf{model}} I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})}$$

- Distribution matching with CI regularization
- Symmetric KL divergence $D_{\mathsf{sym}}(p, q) \triangleq D_{\mathsf{KL}}(p \,\|\, q) + D_{\mathsf{KL}}(q \,\|\, p)$
- Variational density-ratio estimation technique [Pu+17]
- Auxiliary losses: reconstruction losses, latent-matching losses, cross-matching loss, ...

Training method

- For each model $p_{\mathsf{model}} \in \{p_{\to\mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$,

$$\boxed{\text{minimize} \quad D(p_{\mathsf{model}}, q_{\mathsf{xy}\to}) + \lambda_{\mathsf{model}}^{\mathsf{CI}} I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})}$$

- Distribution matching with CI regularization
- Symmetric KL divergence $D_{\mathsf{sym}}(p, q) \triangleq D_{\mathsf{KL}}(p \parallel q) + D_{\mathsf{KL}}(q \parallel p)$
- Variational density-ratio estimation technique [Pu+17]
- Auxiliary losses: reconstruction losses, latent-matching losses, cross-matching loss, ...
- Simultaneous training: minimize a weighted sum of the objectives

Training method

- For each model $p_{\text{model}} \in \{p_{\to\text{xy}}, p_{\text{x}\to\text{y}}, p_{\text{y}\to\text{x}}\}$,

$$\boxed{\text{minimize} \quad D(p_{\text{model}}, q_{\text{xy}\to}) + \lambda_{\text{model}}^{\text{CI}} I_{\text{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})}$$

- Distribution matching with CI regularization
- Symmetric KL divergence $D_{\text{sym}}(p, q) \triangleq D_{\text{KL}}(p \parallel q) + D_{\text{KL}}(q \parallel p)$
- Variational density-ratio estimation technique [Pu+17]
- Auxiliary losses: reconstruction losses, latent-matching losses, cross-matching loss, ...
- Simultaneous training: minimize a weighted sum of the objectives
  - In practice, weights including $\lambda_{\text{model}}^{\text{CI}}$ can be chosen by trial and error

Training method

- For each model $p_{\text{model}} \in \{p_{\to\text{xy}}, p_{\text{x}\to\text{y}}, p_{\text{y}\to\text{x}}\}$,

$$\boxed{\text{minimize} \quad D(p_{\text{model}}, q_{\text{xy}\to}) + \lambda_{\text{model}}^{\text{CI}} I_{\text{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})}$$

- Distribution matching with CI regularization
- Symmetric KL divergence $D_{\text{sym}}(p, q) \triangleq D_{\text{KL}}(p \parallel q) + D_{\text{KL}}(q \parallel p)$
- Variational density-ratio estimation technique [Pu+17]
- Auxiliary losses: reconstruction losses, latent-matching losses, cross-matching loss, ...
- Simultaneous training: minimize a weighted sum of the objectives
- Additional tricks: shared discriminator feature maps, deterministic encoders, instance noise trick

Training method

- For each model $p_{\text{model}} \in \{p_{\to xy}, p_{x\to y}, p_{y\to x}\}$,

$$\boxed{\text{minimize} \quad D(p_{\text{model}}, q_{xy\to}) + \lambda_{\text{model}}^{\text{CI}} I_{\text{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})}$$

- Distribution matching with CI regularization
- Symmetric KL divergence $D_{\text{sym}}(p, q) \triangleq D_{\text{KL}}(p \parallel q) + D_{\text{KL}}(q \parallel p)$
- Variational density-ratio estimation technique [Pu+17]
- Auxiliary losses: reconstruction losses, latent-matching losses, cross-matching loss, ...
- Simultaneous training: minimize a weighted sum of the objectives
- Additional tricks: shared discriminator feature maps, deterministic encoders, instance noise trick
- Plug-in deep neural networks for encoders, decoders, discriminators

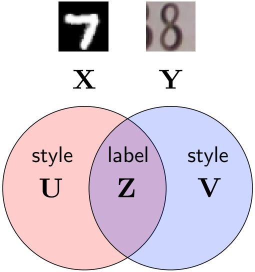
Experiment. MNIST–SVHN add-1 dataset

- $(\mathbf{X}, \mathbf{Y})$=(MNIST, SVHN) with label(SVHN)=label(MNIST)+1

Experiment. MNIST–SVHN add-1 dataset

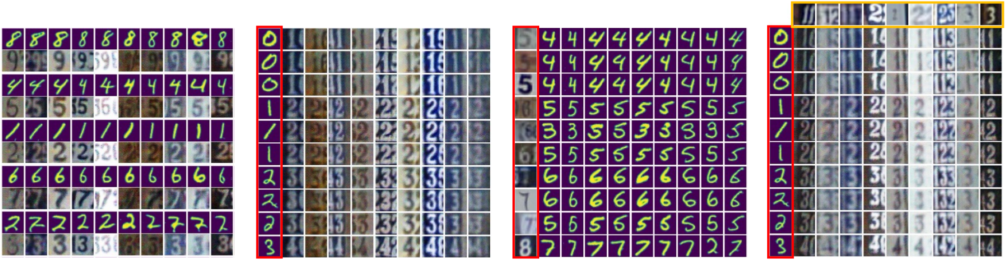
- $(\mathbf{X}, \mathbf{Y})$=(MNIST, SVHN) with label(SVHN)=label(MNIST)+1

- $\mathbf{Z}$=label, $(\mathbf{U}, \mathbf{V}) \approx$ (style of MNIST, style of SVHN)



Experiment. MNIST–SVHN add-1 dataset

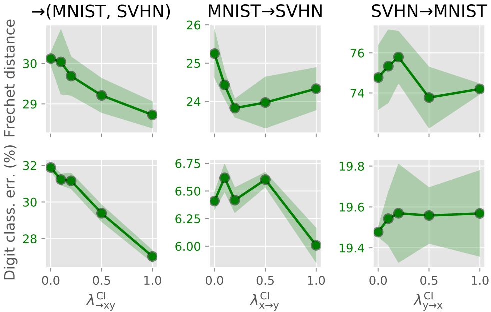
- Generated samples: same $\mathbf{z}$ across the rows; same $\mathbf{u}, \mathbf{v}$ across the columns
- A red box highlights inputs; a yellow box highlight style references



(a) →(MNIST,SVHN) (b) MNIST→SVHN (c) SVHN→MNIST (d) MNIST→SVHN with style transfer

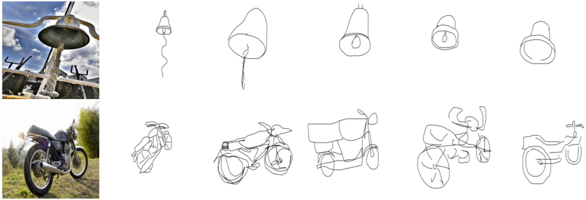
Experiment. MNIST–SVHN add-1 dataset

- Numerical evaluation: $\lambda^{\mathsf{CI}}_{\mathsf{model}}$ vs. quality of generated samples
- Frechet distance: measures a distance between generated samples and test dataset
- Digit classification error: computed by pretrained MNIST/SVHN classifiers



Experiment. Sketchy dataset [San+16]

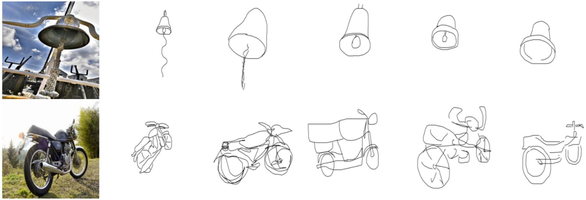
- $(\mathbf{X}, \mathbf{Y})$=(photo, human sketch)



- $\mathbf{Z} \approx$ image class, $(\mathbf{U}, \mathbf{V}) \approx$ (variation in photo, style of sketch)

Experiment. Sketchy dataset [San+16]

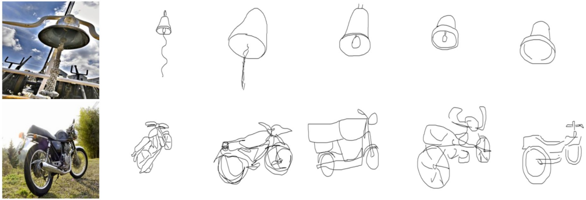
- $(\mathbf{X}, \mathbf{Y})$=(photo, human sketch)



- $\mathbf{Z} \approx$ image class, $(\mathbf{U}, \mathbf{V}) \approx$ (variation in photo, style of sketch)
- Cross-domain retrieval: given a sketch $(\mathbf{y})$, retrieve photos $(\mathbf{x})$

Experiment. Sketchy dataset [San+16]

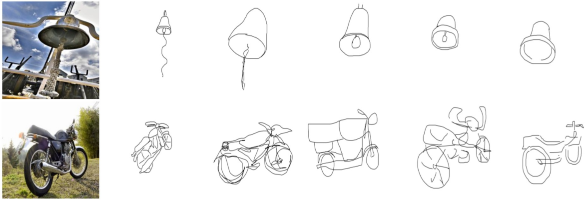
- $(\mathbf{X}, \mathbf{Y})$=(photo, human sketch)



- $\mathbf{Z} \approx$ image class, $(\mathbf{U}, \mathbf{V}) \approx$ (variation in photo, style of sketch)
- Cross-domain retrieval: given a sketch $(\mathbf{y})$, retrieve photos $(\mathbf{x})$
- Our method: retrieve via common representations

Experiment. Sketchy dataset [San+16]

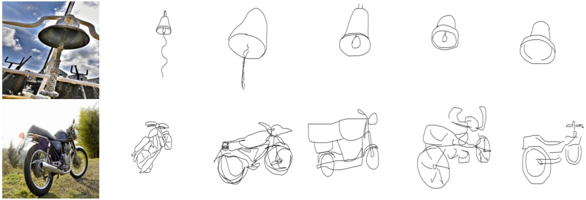
- $(\mathbf{X}, \mathbf{Y})$=(photo, human sketch)



- $\mathbf{Z} \approx$ image class, $(\mathbf{U}, \mathbf{V}) \approx$ (variation in photo, style of sketch)
- Cross-domain retrieval: given a sketch $(\mathbf{y})$, retrieve photos $(\mathbf{x})$
- Our method: retrieve via common representations
    - Train both conditional models;

Experiment. Sketchy dataset [San+16]

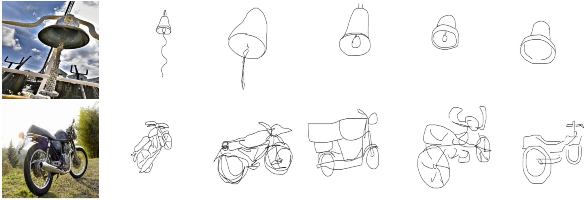
- $(\mathbf{X}, \mathbf{Y})$=(photo, human sketch)



- $\mathbf{Z} \approx$ image class, $(\mathbf{U}, \mathbf{V}) \approx$ (variation in photo, style of sketch)
- Cross-domain retrieval: given a sketch $(\mathbf{y})$, retrieve photos $(\mathbf{x})$
- Our method: retrieve via common representations
  - Train both conditional models;
  - Using $q_\theta(\mathbf{z}|\mathbf{x})$, register common representations $\{\mathbf{z}_j\}_{j\in[n]}$ of test photos $\{\mathbf{x}_j\}_{j\in[n]}$;

Experiment. Sketchy dataset [San+16]

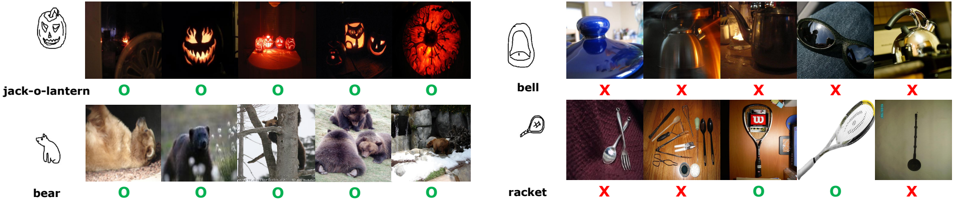
- $(\mathbf{X}, \mathbf{Y})$=(photo, human sketch)



- $\mathbf{Z} \approx$ image class, $(\mathbf{U}, \mathbf{V}) \approx$ (variation in photo, style of sketch)
- Cross-domain retrieval: given a sketch $(\mathbf{y})$, retrieve photos $(\mathbf{x})$
- Our method: retrieve via common representations
    - Train both conditional models;
    - Using $q_\theta(\mathbf{z}|\mathbf{x})$, register common representations $\{\mathbf{z}_j\}_{j\in[n]}$ of test photos $\{\mathbf{x}_j\}_{j\in[n]}$;
    - Given a sketch $\mathbf{y}_o$, retrieve the K-nearest neighbors of $\mathbf{z}_o \sim q_\theta(\mathbf{z}|\mathbf{y}_o)$ from $\{\mathbf{z}_j\}_{j\in[n]}$

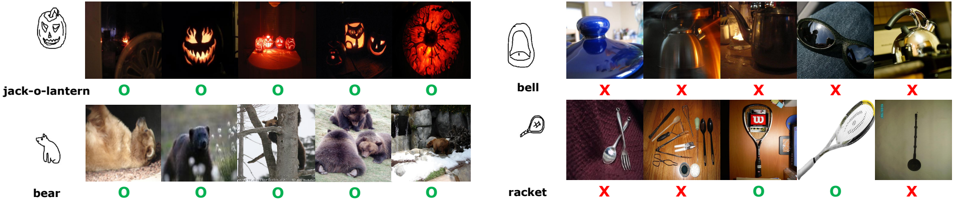
Experiment. Sketchy dataset [San+16]

- Zero-shot: training set has no overlapping classes with test set
- **Examples**: correct retrievals (left) / wrong retrievals (right)



Experiment. Sketchy dataset [San+16]

- Zero-shot: training set has no overlapping classes with test set
- **Examples**: correct retrievals (left) / wrong retrievals (right)



- Numerical evaluation: precision@K (P@K), mean average precision (mAP)

Models	P@100	mAP
LCALE [Lin+20]	0.583	0.476
IIAE [Hwa+20]	0.659	0.573
Variational Wyner	**0.703**	**0.629**

Concluding remarks

- Wyner's common representation:

$$\min_{q(\mathbf{z}|\mathbf{x},\mathbf{y}):\mathbf{X}-\mathbf{Z}-\mathbf{Y}} I(\mathbf{Z};\mathbf{X},\mathbf{Y})$$

Concluding remarks

- Wyner's common representation:

$$\min_{q(\mathbf{z}|\mathbf{x},\mathbf{y}):\mathbf{X}-\mathbf{Z}-\mathbf{Y}} I(\mathbf{Z};\mathbf{X},\mathbf{Y})$$

- Learning distributions with Wyner's common information
  $\rightarrow$ disentangled representations
  $\rightarrow$ better performance in downstream tasks!

Concluding remarks

- Wyner's common representation:

$$\min_{q(\mathbf{z}|\mathbf{x},\mathbf{y}):\mathbf{X}-\mathbf{Z}-\mathbf{Y}} I(\mathbf{Z};\mathbf{X},\mathbf{Y})$$

- Learning distributions with Wyner's common information
  $\rightarrow$ disentangled representations
  $\rightarrow$ better performance in downstream tasks!

Q1. What is the operational meaning of Wyner's common representation?

Concluding remarks

- Wyner's common representation:

$$\min_{q(\mathbf{z}|\mathbf{x},\mathbf{y}):\mathbf{X}-\mathbf{Z}-\mathbf{Y}} I(\mathbf{Z};\mathbf{X},\mathbf{Y})$$

- Learning distributions with Wyner's common information
  $\rightarrow$ disentangled representations
  $\rightarrow$ better performance in downstream tasks!

Q1. What is the operational meaning of Wyner's common representation?

Q2. More than two variables?

Part II

From the Power of Random Guessing to Scalable Nearest-Neighbor Algorithms

Nearest-neighbor classification

- **Data**: Let $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be i.i.d. samples over $\mathcal{X} \times \mathcal{Y}$

Nearest-neighbor classification

- **Data**: Let $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be i.i.d. samples over $\mathcal{X} \times \mathcal{Y}$

  Assume a separable metric space $(\mathcal{X}, \rho)$, e.g., $\mathcal{X} = \mathbb{R}^d$ with Euclidean distance

Nearest-neighbor classification

- **Data**: Let $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be i.i.d. samples over $\mathcal{X} \times \mathcal{Y}$

  Assume a separable metric space $(\mathcal{X}, \rho)$, e.g., $\mathcal{X} = \mathbb{R}^d$ with Euclidean distance

  For binary $\mathcal{Y} = \{0, 1\}$, let $\eta(x) = \mathsf{P}(Y = 1 | X = x)$

Nearest-neighbor classification

- **Data**: Let $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be i.i.d. samples over $\mathcal{X} \times \mathcal{Y}$

  Assume a separable metric space $(\mathcal{X}, \rho)$, e.g., $\mathcal{X} = \mathbb{R}^d$ with Euclidean distance

  For binary $\mathcal{Y} = \{0, 1\}$, let $\eta(x) = \mathsf{P}(Y = 1 | X = x)$
- **Goal**: Construct a classifier $\hat{g} \colon \mathcal{X} \to \mathcal{Y}$ that minimizes $\mathsf{P}\{\hat{g}(X) \neq Y\}$

Nearest-neighbor classification

- **Data**: Let $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be i.i.d. samples over $\mathcal{X} \times \mathcal{Y}$

  Assume a separable metric space $(\mathcal{X}, \rho)$, e.g., $\mathcal{X} = \mathbb{R}^d$ with Euclidean distance

  For binary $\mathcal{Y} = \{0, 1\}$, let $\eta(x) = \mathsf{P}(Y = 1 | X = x)$
- **Goal**: Construct a classifier $\hat{g} \colon \mathcal{X} \to \mathcal{Y}$ that minimizes $\mathsf{P}\{\hat{g}(X) \neq Y\}$
- The Bayes classifier $g^*(x) = 1\{\eta(x) \geq \frac{1}{2}\}$ is optimal

Nearest-neighbor classification

- **Data**: Let $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be i.i.d. samples over $\mathcal{X} \times \mathcal{Y}$

  Assume a separable metric space $(\mathcal{X}, \rho)$, e.g., $\mathcal{X} = \mathbb{R}^d$ with Euclidean distance

  For binary $\mathcal{Y} = \{0, 1\}$, let $\eta(x) = \mathsf{P}(Y = 1|X = x)$
- **Goal**: Construct a classifier $\hat{g}\colon \mathcal{X} \to \mathcal{Y}$ that minimizes $\mathsf{P}\{\hat{g}(X) \neq Y\}$
- The Bayes classifier $g^*(x) = 1\{\eta(x) \geq \frac{1}{2}\}$ is optimal
- k-nearest-neighbor (k-NN) classifier $\hat{g}_{k\text{-NN}}$: for a query x, find the k-nearest neighbors of x and take the majority vote over the labels

Nearest-neighbor classification

- **Data**: Let $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be i.i.d. samples over $\mathcal{X} \times \mathcal{Y}$

  Assume a separable metric space $(\mathcal{X}, \rho)$, e.g., $\mathcal{X} = \mathbb{R}^d$ with Euclidean distance

  For binary $\mathcal{Y} = \{0, 1\}$, let $\eta(x) = \mathsf{P}(Y = 1 | X = x)$
- **Goal**: Construct a classifier $\hat{g} \colon \mathcal{X} \to \mathcal{Y}$ that minimizes $\mathsf{P}\{\hat{g}(X) \neq Y\}$
- The Bayes classifier $g^*(x) = 1\{\eta(x) \geq \frac{1}{2}\}$ is optimal
- k-nearest-neighbor (k-NN) classifier $\hat{g}_{k\text{-NN}}$: for a query x, find the k-nearest neighbors of x and take the majority vote over the labels
- Cover and Hart (1967):

$$\lim_{n \to \infty} \mathsf{P}\{\hat{g}_{\text{1-NN}}(X) \neq Y\} \leq 2\,\mathsf{P}\{g^*(X) \neq Y\}$$

Nearest-neighbor classification

- **Data**: Let $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be i.i.d. samples over $\mathcal{X} \times \mathcal{Y}$

  Assume a separable metric space $(\mathcal{X}, \rho)$, e.g., $\mathcal{X} = \mathbb{R}^d$ with Euclidean distance

  For binary $\mathcal{Y} = \{0, 1\}$, let $\eta(x) = \mathsf{P}(Y = 1 | X = x)$
- **Goal**: Construct a classifier $\hat{g} \colon \mathcal{X} \to \mathcal{Y}$ that minimizes $\mathsf{P}\{\hat{g}(X) \neq Y\}$
- The Bayes classifier $g^*(x) = 1\{\eta(x) \geq \frac{1}{2}\}$ is optimal
- k-nearest-neighbor (k-NN) classifier $\hat{g}_{k\text{-NN}}$: for a query x, find the k-nearest neighbors of x and take the majority vote over the labels
- Cover and Hart (1967):

$$\lim_{n \to \infty} \mathsf{P}\{\hat{g}_{1\text{-NN}}(X) \neq Y\} \leq 2\, \mathsf{P}\{g^*(X) \neq Y\}$$

- Stone (1977): If $k \to \infty$ with $k = o(n)$

$$\lim_{n \to \infty} \mathsf{P}\{\hat{g}_{k\text{-NN}}(X) \neq Y\} = \mathsf{P}\{g^*(X) \neq Y\}$$

Nearest-neighbor algorithms

- Classification, regression, density estimation, density functional estimation, ...

Nearest-neighbor algorithms

- Classification, regression, density estimation, density functional estimation, ...
- (+) Simple, elegant, well-understood

Nearest-neighbor algorithms

- Classification, regression, density estimation, density functional estimation, ...
- $(+)$ Simple, elegant, well-understood
- $(-)$ Not directly applicable for large-scale datasets

Nearest-neighbor algorithms

- Classification, regression, density estimation, density functional estimation, ...
- (+) Simple, elegant, well-understood
- (−) Not directly applicable for large-scale datasets

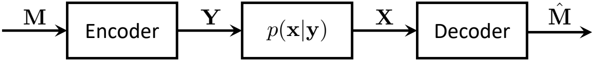
Q. Can we make the k-NN-based algorithms viable in the realm of big data?

Digression: detection problem

- Detect a signal Y from an observation X to minimize $P_e = \mathsf{P}\{\hat{y}(X) \neq Y\}$

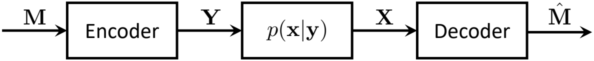
Digression: detection problem

- Detect a signal Y from an observation X to minimize $P_e = \mathsf{P}\{\hat{y}(X) \neq Y\}$
- **Example**: In channel coding, find $\hat{\mathbf{m}}(\mathbf{X})$ that minimizes $\mathsf{P}\{\hat{\mathbf{m}}(\mathbf{X}) \neq \mathbf{M}\}$



Digression: detection problem

- Detect a signal Y from an observation X to minimize $P_e = \mathsf{P}\{\hat{y}(X) \neq Y\}$
- **Example**: In channel coding, find $\hat{\mathbf{m}}(\mathbf{X})$ that minimizes $\mathsf{P}\{\hat{\mathbf{m}}(\mathbf{X}) \neq \mathbf{M}\}$

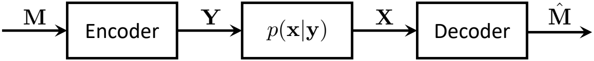


- Maximum a-posteriori probability (MAP) detector:

$$\hat{y}^*(x) = \arg\max_{y \in \mathcal{Y}} p(y|x)$$

Digression: detection problem

- Detect a signal Y from an observation X to minimize $P_e = \mathsf{P}\{\hat{y}(X) \neq Y\}$
- **Example**: In channel coding, find $\hat{\mathbf{m}}(\mathbf{X})$ that minimizes $\mathsf{P}\{\hat{\mathbf{m}}(\mathbf{X}) \neq \mathbf{M}\}$



- Maximum a-posteriori probability (MAP) detector:

$$\hat{y}^*(x) = \arg\max_{y \in \mathcal{Y}} p(y|x)$$

- Randomized likelihood (RL) detector [YAG13]:

$$\hat{Y}(x) \sim p(y|x)$$

Power of random guessing

Liu–Cuff–Verdú lemma (2017)

$$\mathsf{P}\{\hat{Y}(X) \neq Y\} \leq 2P_e^* = 2\,\mathsf{P}\{\hat{y}^*(X) \neq Y\}$$

Power of random guessing

Liu–Cuff–Verdú lemma (2017)

$$\mathsf{P}\{\hat{Y}(X) \neq Y\} \leq 2P_e^* = 2\,\mathsf{P}\{\hat{y}^*(X) \neq Y\}$$

A general factor-of-two bound [Bha+18]

For any metric $d(y, y')$ and $Y \stackrel{d}{=} Y'$,

$$\mathsf{E}[d(Y, Y')] \leq 2 \inf_{y \in \mathcal{Y}} \mathsf{E}[d(Y, y)]$$

Power of random guessing

Liu–Cuff–Verdú lemma (2017)

$$\mathsf{P}\{\hat{Y}(X) \neq Y\} \leq 2P_e^* = 2\,\mathsf{P}\{\hat{y}^*(X) \neq Y\}$$

A general factor-of-two bound [Bha+18]

For any metric $d(y, y')$ and $Y \stackrel{d}{=} Y'$,

$$\mathsf{E}[d(Y, Y')] \leq 2 \inf_{y \in \mathcal{Y}} \mathsf{E}[d(Y, y)]$$

- *Proof.* Triangle inequality

Power of random guessing

Liu–Cuff–Verdú lemma (2017)

$$\mathsf{P}\{\hat{Y}(X) \neq Y\} \leq 2P_e^* = 2\,\mathsf{P}\{\hat{y}^*(X) \neq Y\}$$

A general factor-of-two bound [Bha+18]

For any metric $d(y, y')$ and $Y \stackrel{d}{=} Y'$,

$$\mathsf{E}[d(Y, Y')] \leq 2 \inf_{y \in \mathcal{Y}} \mathsf{E}[d(Y, y)]$$

- *Proof.* Triangle inequality
- *Proof of the LCV lemma.* Let $d(y, \hat{y}) = \mathbb{1}\{y \neq y'\}$, apply the general bound for each x, and take expectation w.r.t. X

Power of random guessing and 1-NN classifier

- Cover and Hart (1967):

$$\lim_{n\to\infty} \mathsf{P}\{\hat{g}_{\text{1-NN}}(X) \neq Y\} \leq 2\,\mathsf{P}\{g^*(X) \neq Y\}$$

Power of random guessing and 1-NN classifier

- Cover and Hart (1967):

$$\lim_{n\to\infty} \mathsf{P}\{\hat{g}_{\mathsf{1\text{-}NN}}(X) \neq Y\} \leq 2\,\mathsf{P}\{g^*(X) \neq Y\}$$

can be thought as a manifestation of the power of random guessing

Power of random guessing and 1-NN classifier

- Cover and Hart (1967):

$$\lim_{n\to\infty} \mathsf{P}\{\hat{g}_{\text{1-NN}}(X) \neq Y\} \leq 2\,\mathsf{P}\{g^*(X) \neq Y\}$$

can be thought as a manifestation of the power of random guessing

Lemma (Cover and Hart, 1967)

Let $X_{(1)}(x)$ be the nearest neighbor of x from i.i.d. samples $\{X_1, \ldots, X_n\}$
If $(\mathcal{X}, \rho)$ is a separable metric space,

$$\lim_{n\to\infty} \rho(X_{(1)}(x), x) = 0 \text{ with probability } 1$$

Power of random guessing and 1-NN classifier

- Cover and Hart (1967):

$$\lim_{n\to\infty} \mathsf{P}\{\hat{g}_{\text{1-NN}}(X) \neq Y\} \leq 2\,\mathsf{P}\{g^*(X) \neq Y\}$$

can be thought as a manifestation of the power of random guessing

Lemma (Cover and Hart, 1967)

Let $X_{(1)}(x)$ be the nearest neighbor of x from i.i.d. samples $\{X_1, \ldots, X_n\}$
If $(\mathcal{X}, \rho)$ is a separable metric space,

$$\lim_{n\to\infty} \rho(X_{(1)}(x), x) = 0 \text{ with probability } 1$$

- Observation 1. (1-NN classifier $\equiv$ RL detector) in the sample limit

Power of multiple random guessing

- Let $\{Y_1'(x), \ldots, Y_M'(x)\}$ be a set of conditionally i.i.d. copies of $Y|\{X = x\}$ and

$$\hat{Y}_M(x) = \mathrm{mode}(Y_1'(x), \ldots, Y_M'(x))$$

Power of multiple random guessing

- Let $\{Y_1'(x), \ldots, Y_M'(x)\}$ be a set of conditionally i.i.d. copies of $Y|\{X = x\}$ and

$$\hat{Y}_M(x) = \mathrm{mode}(Y_1'(x), \ldots, Y_M'(x))$$

Theorem [Bha+18]

For any $\delta > 0$

$$\mathsf{P}\{\hat{Y}_M(X) \neq Y\} \leq P_e^* + O(M)(e^{-\delta^2 \Omega(M)} + \mathsf{P}\{\Delta(X) \leq \delta\})$$

where

$$\Delta(x) \triangleq (\text{the gap between the first and second largest values of } \{p(y|x)\}_{y \in \mathcal{Y}})$$

Power of multiple random guessing

- Let $\{Y_1'(x), \ldots, Y_M'(x)\}$ be a set of conditionally i.i.d. copies of $Y|\{X = x\}$ and

$$\hat{Y}_M(x) = \mathrm{mode}(Y_1'(x), \ldots, Y_M'(x))$$

Theorem [Bha+18]

For any $\delta > 0$

$$\mathsf{P}\{\hat{Y}_M(X) \neq Y\} \leq P_e^* + O(M)(e^{-\delta^2 \Omega(M)} + \mathsf{P}\{\Delta(X) \leq \delta\})$$

where

$$\Delta(x) \triangleq (\text{the gap between the first and second largest values of } \{p(y|x)\}_{y \in \mathcal{Y}})$$

- *Proof.* Hoeffding and Vapnik–Chervonenkis

Power of multiple random guessing

- Let $\{Y_1'(x), \ldots, Y_M'(x)\}$ be a set of conditionally i.i.d. copies of $Y|\{X = x\}$ and

$$\hat{Y}_M(x) = \mathrm{mode}(Y_1'(x), \ldots, Y_M'(x))$$

Theorem [Bha+18]

For any $\delta > 0$

$$\mathsf{P}\{\hat{Y}_M(X) \neq Y\} \leq P_e^* + O(M)(e^{-\delta^2 \Omega(M)} + \mathsf{P}\{\Delta(X) \leq \delta\})$$

where

$$\Delta(x) \triangleq \text{(the gap between the first and second largest values of } \{p(y|x)\}_{y \in \mathcal{Y}})$$

- *Proof.* Hoeffding and Vapnik–Chervonenkis
- Observation 2. (majority vote over M random guesses $\to$ MAP detector) as $M \to \infty$

Power of multiple random guessing with 1-NN classifier

- Observation 1. (1-NN classifier $\equiv$ RL detector) in the sample limit
- Observation 2. (majority vote over M random guesses $\to$ MAP detector) as $M \to \infty$

Power of multiple random guessing with 1-NN classifier

- Observation 1. (1-NN classifier $\equiv$ RL detector) in the sample limit
- Observation 2. (majority vote over M random guesses $\to$ MAP detector) as $M \to \infty$
- The M-NN classifier is one way to emulate the power of multiple random guessing

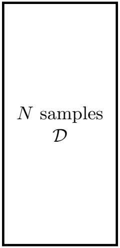
$$\hat{g}_{M\text{-NN}}(x) = \mathrm{mode}(Y_{(1)}(x), \ldots, Y_{(M)}(x))$$

Power of multiple random guessing with 1-NN classifier

- Observation 1. (1-NN classifier $\equiv$ RL detector) in the sample limit
- Observation 2. (majority vote over M random guesses $\to$ MAP detector) as $M \to \infty$
- **Proposal**: aggregate multiple 1-NN classifiers with sample splitting

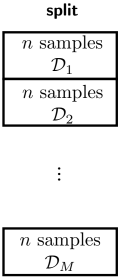
Power of multiple random guessing with 1-NN classifier

- Observation 1. (1-NN classifier $\equiv$ RL detector) in the sample limit
- Observation 2. (majority vote over M random guesses $\to$ MAP detector) as $M \to \infty$
- **Proposal**: aggregate multiple 1-NN classifiers with sample splitting



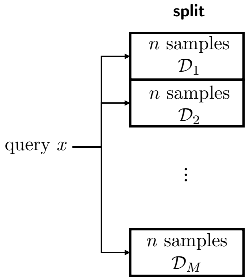
Power of multiple random guessing with 1-NN classifier

- Observation 1. (1-NN classifier $\equiv$ RL detector) in the sample limit
- Observation 2. (majority vote over M random guesses $\to$ MAP detector) as $M \to \infty$
- **Proposal**: aggregate multiple 1-NN classifiers with sample splitting



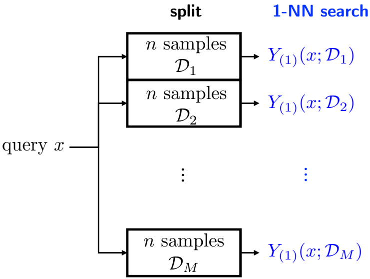
Power of multiple random guessing with 1-NN classifier

- Observation 1. (1-NN classifier $\equiv$ RL detector) in the sample limit
- Observation 2. (majority vote over M random guesses $\to$ MAP detector) as $M \to \infty$
- **Proposal**: aggregate multiple 1-NN classifiers with sample splitting



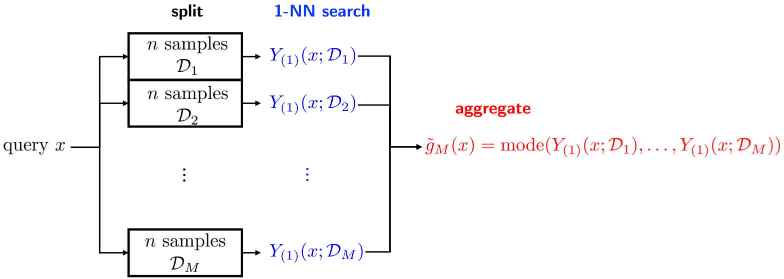
Power of multiple random guessing with 1-NN classifier

- Observation 1. (1-NN classifier $\equiv$ RL detector) in the sample limit
- Observation 2. (majority vote over M random guesses $\to$ MAP detector) as $M \to \infty$
- **Proposal**: aggregate multiple 1-NN classifiers with sample splitting



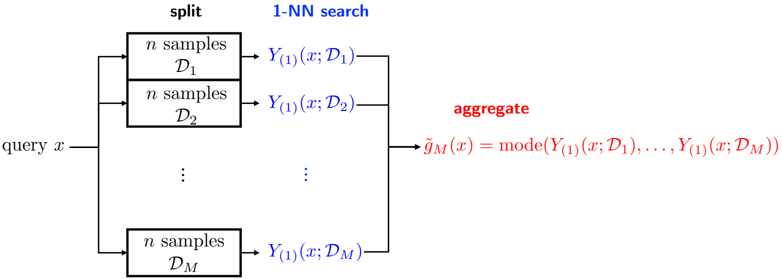
Power of multiple random guessing with 1-NN classifier

- Observation 1. (1-NN classifier $\equiv$ RL detector) in the sample limit
- Observation 2. (majority vote over M random guesses $\to$ MAP detector) as $M \to \infty$
- **Proposal**: aggregate multiple 1-NN classifiers with sample splitting



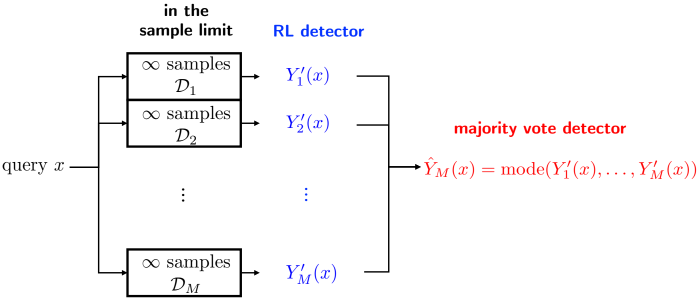
Power of multiple random guessing with 1-NN classifier

- Observation 1. (1-NN classifier $\equiv$ RL detector) in the sample limit
- Observation 2. (majority vote over M random guesses $\to$ MAP detector) as $M \to \infty$
- **Proposal**: aggregate multiple 1-NN classifiers with sample splitting
- We call the resulting classifier $\tilde{g}_M$ the M-split 1-NN classifier



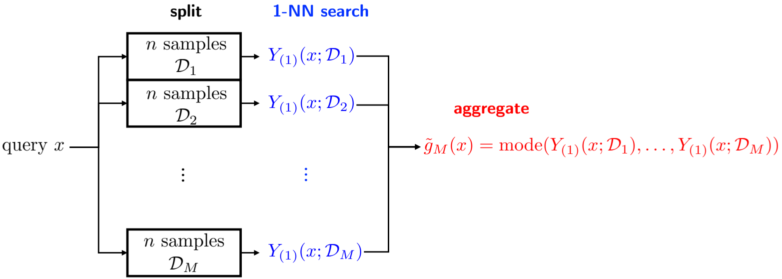
Power of multiple random guessing with 1-NN classifier

- Observation 1. (1-NN classifier $\equiv$ RL detector) in the sample limit
- Observation 2. (majority vote over M random guesses $\to$ MAP detector) as $M \to \infty$
- **Proposal**: aggregate multiple 1-NN classifiers with sample splitting
- We call the resulting classifier $\tilde{g}_M$ the M-split 1-NN classifier



Power of multiple random guessing with 1-NN classifier

- Observation 1. (1-NN classifier $\equiv$ RL detector) in the sample limit
- Observation 2. (majority vote over M random guesses $\to$ MAP detector) as $M \to \infty$
- **Proposal**: aggregate multiple 1-NN classifiers with sample splitting
- We call the resulting classifier $\tilde{g}_M$ the M-split 1-NN classifier
- Fully parallelizable; with S workers, query complexity becomes $1/S$



Performance guarantee

Theorem (excess risk) [RK22]

For $\mathcal{X} = \mathbb{R}^d$ with metric $\rho(x, x')$, assume:

Performance guarantee

Theorem (excess risk) [RK22]

For $\mathcal{X} = \mathbb{R}^d$ with metric $\rho(x, x')$, assume:

1. $\eta(x) = \mathsf{P}\{Y = 1 | X = x\}$ is (α, A)-*Hölder continuous* for some $0 < \alpha \leq 1$ and $A > 0$, i.e., $\forall x, x' \in \mathcal{X}$,

$$|\eta(x) - \eta(x')| \leq A\rho^{\alpha}(x, x').$$

Performance guarantee

Theorem (excess risk) [RK22]

For $\mathcal{X} = \mathbb{R}^d$ with metric $\rho(x, x')$, assume:

1. $\eta(x) = \mathsf{P}\{Y = 1|X = x\}$ is *(α, A)-Hölder continuous* for some $0 < \alpha \leq 1$ and $A > 0$, i.e., $\forall x, x' \in \mathcal{X}$,

$$|\eta(x) - \eta(x')| \leq A\rho^{\alpha}(x, x').$$

2. η satisfies the *β-margin condition* for $\beta > 0$, i.e., $\exists\ C > 0$ s.t.

$$\mathsf{P}\left\{\left|\eta(X) - \frac{1}{2}\right| \leq \Delta\right\} \leq C\Delta^{\beta}$$

Performance guarantee

Theorem (excess risk) [RK22]

For $\mathcal{X} = \mathbb{R}^d$ with metric $\rho(x, x')$, assume:

1. $\eta(x) = \mathsf{P}\{Y = 1 | X = x\}$ is *(α, A)-Hölder continuous* for some $0 < \alpha \le 1$ and $A > 0$, i.e., $\forall x, x' \in \mathcal{X}$,

$$|\eta(x) - \eta(x')| \le A\rho^\alpha(x, x').$$

2. η satisfies the *β-margin condition* for $\beta > 0$, i.e., $\exists\ C > 0$ s.t.

$$\mathsf{P}\left\{\left|\eta(X) - \frac{1}{2}\right| \le \Delta\right\} \le C\Delta^\beta$$

For $M = \Theta(N^{\frac{2\alpha}{2\alpha+d}})$, $\mathsf{E}[\mathsf{P}\{\tilde{g}_M(X) \ne Y\}] - \mathsf{P}\{g^*(X) \ne Y\} = \tilde{O}(N^{-\frac{(\beta+1)\alpha}{2\alpha+d}})$

Performance guarantee

Theorem (excess risk) [RK22]

For $\mathcal{X} = \mathbb{R}^d$ with metric $\rho(x, x')$, assume:

1. $\eta(x) = \mathsf{P}\{Y = 1 | X = x\}$ is *(α, A)-Hölder continuous* for some $0 < \alpha \leq 1$ and $A > 0$, i.e., $\forall x, x' \in \mathcal{X}$,
$$|\eta(x) - \eta(x')| \leq A\rho^\alpha(x, x').$$
2. η satisfies the *β-margin condition* for $\beta > 0$, i.e., $\exists\ C > 0$ s.t.
$$\mathsf{P}\left\{\left|\eta(X) - \frac{1}{2}\right| \leq \Delta\right\} \leq C\Delta^\beta$$

For $M = \Theta(N^{\frac{2\alpha}{2\alpha+d}})$, $\mathsf{E}[\mathsf{P}\{\tilde{g}_M(X) \neq Y\}] - \mathsf{P}\{g^*(X) \neq Y\} = \tilde{O}(N^{-\frac{(\beta+1)\alpha}{2\alpha+d}})$

- Nearly minimax-optimal [AT+07]

Performance guarantee

Theorem (excess risk) [RK22]

For $\mathcal{X} = \mathbb{R}^d$ with metric $\rho(x, x')$, assume:

1. $\eta(x) = \mathsf{P}\{Y = 1 | X = x\}$ is *(α, A)-Hölder continuous* for some $0 < \alpha \le 1$ and $A > 0$, i.e., $\forall x, x' \in \mathcal{X}$,
$$|\eta(x) - \eta(x')| \le A\rho^\alpha(x, x').$$
2. η satisfies the *β-margin condition* for $\beta > 0$, i.e., $\exists\ C > 0$ s.t.
$$\mathsf{P}\left\{\left|\eta(X) - \frac{1}{2}\right| \le \Delta\right\} \le C\Delta^\beta$$

For $M = \Theta(N^{\frac{2\alpha}{2\alpha+d}})$, $\mathsf{E}[\mathsf{P}\{\tilde{g}_M(X) \neq Y\}] - \mathsf{P}\{g^*(X) \neq Y\} = \tilde{O}(N^{-\frac{(\beta+1)\alpha}{2\alpha+d}})$

- Nearly minimax-optimal [AT+07]
- The M-split 1-NN classifier emulates a $\Theta(M)$-NN classifier [CD14]

Performance guarantee

Theorem (excess risk) [RK22]

For $\mathcal{X} = \mathbb{R}^d$ with metric $\rho(x, x')$, assume:

1. $\eta(x) = \mathsf{P}\{Y = 1 | X = x\}$ is *(α, A)-Hölder continuous* for some $0 < \alpha \le 1$ and $A > 0$, i.e., $\forall x, x' \in \mathcal{X}$,

$$|\eta(x) - \eta(x')| \le A\rho^\alpha(x, x').$$

2. η satisfies the *β-margin condition* for $\beta > 0$, i.e., $\exists\ C > 0$ s.t.

$$\mathsf{P}\left\{\left|\eta(X) - \frac{1}{2}\right| \le \Delta\right\} \le C\Delta^\beta$$

For $M = \Theta(N^{\frac{2\alpha}{2\alpha+d}})$, $\mathsf{E}[\mathsf{P}\{\tilde{g}_M(X) \neq Y\}] - \mathsf{P}\{g^*(X) \neq Y\} = \tilde{O}(N^{-\frac{(\beta+1)\alpha}{2\alpha+d}})$

- Nearly minimax-optimal [AT+07]
- The M-split 1-NN classifier emulates a $\Theta(M)$-NN classifier [CD14]
- Proof idea: analyze an intermediate distance-selective rule

Concluding remarks

- An existing divide-and-conquer framework [QDC19] requires $k \to \infty$ for the base k-NN classifier, to be optimal

Concluding remarks

- An existing divide-and-conquer framework [QDC19] requires $k \to \infty$ for the base k-NN classifier, to be optimal
- Aggregating multiple runs of the simplest 1-NN search is all we need!

Concluding remarks

- An existing divide-and-conquer framework [QDC19] requires $k \to \infty$ for the base k-NN classifier, to be optimal
- Aggregating multiple runs of the simplest 1-NN search is all we need!
- cf. distributed NN search [FMP20], approximate NN search [HIM12]

Concluding remarks

- An existing divide-and-conquer framework [QDC19] requires $k \to \infty$ for the base k-NN classifier, to be optimal
- Aggregating multiple runs of the simplest 1-NN search is all we need!
- cf. distributed NN search [FMP20], approximate NN search [HIM12]
- The same framework works for regression and can be extended to density estimation

Concluding remarks

- An existing divide-and-conquer framework [QDC19] requires $k \to \infty$ for the base k-NN classifier, to be optimal
- Aggregating multiple runs of the simplest 1-NN search is all we need!
- cf. distributed NN search [FMP20], approximate NN search [HIM12]
- The same framework works for regression and can be extended to density estimation

Q. Split-and-aggregate framework for other nonparametric algorithms?

Acknowledgments

- **My advisors**: Prof. Young-Han Kim and Prof. Sanjoy Dasgupta

Acknowledgments

- **My advisors**: Prof. Young-Han Kim and Prof. Sanjoy Dasgupta
- **Committee members**:
  Prof. Ery Arias-Castro, Prof. Yoav Freund, Prof. Nikolay Atanasov, Prof. Piya Pal

Acknowledgments

- **My advisors**: Prof. Young-Han Kim and Prof. Sanjoy Dasgupta
- **Committee members**:
  Prof. Ery Arias-Castro, Prof. Yoav Freund, Prof. Nikolay Atanasov, Prof. Piya Pal
- **Funding sources**: NAVER, Samsung, NSF, Kwanjeong educational foundation

Acknowledgments

- **My advisors**: Prof. Young-Han Kim and Prof. Sanjoy Dasgupta
- **Committee members**:
  Prof. Ery Arias-Castro, Prof. Yoav Freund, Prof. Nikolay Atanasov, Prof. Piya Pal
- **Funding sources**: NAVER, Samsung, NSF, Kwanjeong educational foundation
- **Internship mentors**: Dr. Yoojin Choi (Samsung), Dr. Yang Yang (Qualcomm)

Acknowledgments

- **My advisors**: Prof. Young-Han Kim and Prof. Sanjoy Dasgupta
- **Committee members**:
  Prof. Ery Arias-Castro, Prof. Yoav Freund, Prof. Nikolay Atanasov, Prof. Piya Pal
- **Funding sources**: NAVER, Samsung, NSF, Kwanjeong educational foundation
- **Internship mentors**: Dr. Yoojin Choi (Samsung), Dr. Yang Yang (Qualcomm)
- **Prof. Kim's group members**

Acknowledgments

- **My advisors**: Prof. Young-Han Kim and Prof. Sanjoy Dasgupta
- **Committee members**:
  Prof. Ery Arias-Castro, Prof. Yoav Freund, Prof. Nikolay Atanasov, Prof. Piya Pal
- **Funding sources**: NAVER, Samsung, NSF, Kwanjeong educational foundation
- **Internship mentors**: Dr. Yoojin Choi (Samsung), Dr. Yang Yang (Qualcomm)
- **Prof. Kim's group members**
- **Prof. Dasgupta's group members**

Acknowledgments

- **My advisors**: Prof. Young-Han Kim and Prof. Sanjoy Dasgupta
- **Committee members**:
  Prof. Ery Arias-Castro, Prof. Yoav Freund, Prof. Nikolay Atanasov, Prof. Piya Pal
- **Funding sources**: NAVER, Samsung, NSF, Kwanjeong educational foundation
- **Internship mentors**: Dr. Yoojin Choi (Samsung), Dr. Yang Yang (Qualcomm)
- **Prof. Kim's group members**
- **Prof. Dasgupta's group members**
- **My friends**

Acknowledgments

- **My advisors**: Prof. Young-Han Kim and Prof. Sanjoy Dasgupta
- **Committee members**:
  Prof. Ery Arias-Castro, Prof. Yoav Freund, Prof. Nikolay Atanasov, Prof. Piya Pal
- **Funding sources**: NAVER, Samsung, NSF, Kwanjeong educational foundation
- **Internship mentors**: Dr. Yoojin Choi (Samsung), Dr. Yang Yang (Qualcomm)
- **Prof. Kim's group members**
- **Prof. Dasgupta's group members**
- **My friends**
- **My parents**

Acknowledgments

- **My advisors**: Prof. Young-Han Kim and Prof. Sanjoy Dasgupta
- **Committee members**:
  Prof. Ery Arias-Castro, Prof. Yoav Freund, Prof. Nikolay Atanasov, Prof. Piya Pal
- **Funding sources**: NAVER, Samsung, NSF, Kwanjeong educational foundation
- **Internship mentors**: Dr. Yoojin Choi (Samsung), Dr. Yang Yang (Qualcomm)
- **Prof. Kim's group members**
- **Prof. Dasgupta's group members**
- **My friends**
- **My parents**
- **My wife** Kyungeun

Acknowledgments

- **My advisors**: Prof. Young-Han Kim and Prof. Sanjoy Dasgupta
- **Committee members**:
  Prof. Ery Arias-Castro, Prof. Yoav Freund, Prof. Nikolay Atanasov, Prof. Piya Pal
- **Funding sources**: NAVER, Samsung, NSF, Kwanjeong educational foundation
- **Internship mentors**: Dr. Yoojin Choi (Samsung), Dr. Yang Yang (Qualcomm)
- **Prof. Kim's group members**
- **Prof. Dasgupta's group members**
- **My friends**
- **My parents**
- **My wife** Kyungeun
- **My babies** Arielle and Asher

Thank you!

References I

(* and † indicate equal contribution and alphabetical ordering, respectively.)

[AT+07] J.-Y. Audibert, A. B. Tsybakov, et al. "Fast learning rates for plug-in classifiers". In: *Ann. Stat.* 35.2 (2007), pp. 608–633.

[BRK22] A. Bhatt*, **J. J. Ryu***, and Y.-H. Kim. "On Universal Portfolios with Continuous Side Information". In: (2022). arXiv: 2202.02431 [cs.IT].

[Bha+18] A. Bhatt†, J.-T. Huang†, Y.-H. Kim†, **J. J. Ryu**†, and P. Sen†. "Variations on a theme by Liu, Cuff, and Verdú: The power of posterior sampling". In: *Proc. IEEE Inf. Theory Workshop*. IEEE. 2018, pp. 1–5.

[CD14] K. Chaudhuri and S. Dasgupta. "Rates of convergence for nearest neighbor classification". In: *Adv. Neural Info. Proc. Syst.* Vol. 27. Curran Associates, Inc., 2014, pp. 3437–3445.

[CH67] T. M. Cover and P. Hart. "Nearest neighbor pattern classification". In: *IEEE Trans. Inf. Theory* 13.1 (1967), pp. 21–27.

References II

[FMP20] R. Fathi, A. R. Molla, and G. Pandurangan. "Efficient distributed algorithms for the k-nearest neighbors problem". In: *Proc. 32nd ACM Symp. Parallelism Algorithms Archit.* 2020, pp. 527–529.

[HIM12] S. Har-Peled, P. Indyk, and R. Motwani. "Approximate nearest neighbor: Towards removing the curse of dimensionality". In: *Theory Comput.* 8.1 (2012), pp. 321–350.

[Hwa+20] H. Hwang, G.-H. Kim, S. Hong, and K.-E. Kim. "Variational Interaction Information Maximization for Cross-domain Disentanglement". In: *Adv. Neural Info. Proc. Syst.* Vol. 33. 2020.

[Lin+20] K. Lin, X. Xu, L. Gao, Z. Wang, and H. T. Shen. "Learning cross-aligned latent embeddings for zero-shot cross-modal retrieval". In: *Proc. AAAI Conf. Artif. Int.* Vol. 34. 2020, pp. 11515–11522.

[LCV17] J. Liu, P. Cuff, and S. Verdú. "On α-decodability and α-likelihood decoder". In: *Proc. 55th Ann. Allerton Conf. Comm. Control Comput.* Monticello, IL, Oct. 2017.

References III

[Pu+17] Y. Pu, W. Wang, R. Henao, L. Chen, Z. Gan, C. Li, and L. Carin. "Adversarial symmetric variational autoencoder". In: *Adv. Neural Info. Proc. Syst.* 2017, pp. 4330–4339.

[QDC19] X. Qiao, J. Duan, and G. Cheng. "Rates of Convergence for Large-scale Nearest Neighbor Classification". In: *Adv. Neural Info. Proc. Syst.* Vol. 32. Curran Associates, Inc., 2019, pp. 10768–10779.

[RBK22] **J. J. Ryu**, A. Bhatt, and Y.-H. Kim. "Parameter-Free Online Linear Optimization with Side Information via Universal Coin Betting". In: *Int. Conf. Artif. Int. Stat.* 2022. arXiv: 2202.02431 [cs.IT].

[Ryu+21] **J. J. Ryu**, Y. Choi, Y.-H. Kim, M. El-Khamy, and J. Lee. "Learning with Succinct Common Representation Based on Wyner's Common Information". In: (2021). *In preparation*; An extended abstract was presented in Bayesian Deep Learning Workshop at NeurIPS in 2021.

References IV

[Ryu+22] **J. J. Ryu***, S. Ganguly*, Y.-H. Kim, Y.-K. Noh, and D. D. Lee. "Nearest neighbor density functional estimation from inverse Laplace transform". In: *IEEE Trans. Inf. Theory* (2022). to appear. arXiv: 1805.08342 [math.ST].

[RHK21] **J. J. Ryu**, J.-T. Huang, and Y.-H. Kim. "On the Role of Eigendecomposition in Kernel Embedding". In: *Proc. IEEE Int. Symp. Inf. Theory*. IEEE. 2021, pp. 2030–2035.

[RK22] **J. J. Ryu** and Y.-H. Kim. "One-Nearest-Neighbor Search Is All You Need for Minimax Regression and Classification". 2022. arXiv: 2202.02464 [math.ST].

[RK18] **J. Ryu** and Y.-H. Kim. "Conditional distribution learning with neural networks and its application to universal image denoising". In: *Proc. IEEE Int. Conf. Image Proc.* IEEE. 2018, pp. 3214–3218.

References V

[San+16] P. Sangkloy, N. Burnell, C. Ham, and J. Hays. "The Sketchy Database: Learning to Retrieve Badly Drawn Bunnies". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* (2016).

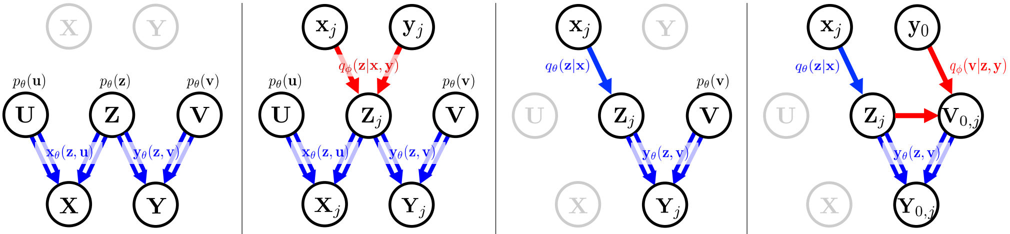
[Shi+19] Y. Shi, N. Siddharth, B. Paige, and P. H. Torr. "Variational mixture-of-experts autoencoders for multi-modal deep generative models". In: *Adv. Neural Info. Proc. Syst.* Vol. 32. 2019.

[Sto77] C. J. Stone. "Consistent nonparametric regression". In: *Ann. Stat.* (1977), pp. 595–620.

[YAG13] M. H. Yassaee, M. R. Aref, and A. Gohari. "A technique for deriving one-shot achievability results in network information theory". In: *Proc. IEEE Int. Symp. Inf. Theory*. Istanbul, Turkey, 2013, pp. 1151–1155.

Backup Slides

How to use the variational Wyner model



(a) Joint sampling

(b) Joint stochastic reconstruction

(c) Conditional sampling

(d) Conditional sampling with style control

- Variational encoders are introduced for training, but can be also used in sampling
- Local variational encoders $q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x}), q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y})$ can be viewed as style extractors

Derivation

- For each model $p_{\mathsf{model}} \in \{p_{\to\mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$:

$$\boxed{\begin{array}{ll} \text{minimize} & I_{\mathsf{xy}\to}(\mathbf{X},\mathbf{Y};\mathbf{Z}) \\ \text{subject to} & \mathbf{X} - \mathbf{Z} - \mathbf{Y} \\ \text{variables} & q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \end{array}}$$

Derivation

- For each model $p_{\text{model}} \in \{p_{\to\text{xy}}, p_{\text{x}\to\text{y}}, p_{\text{y}\to\text{x}}\}$:

$$\boxed{\begin{array}{ll} \text{minimize} & I_{\text{xy}\to}(\mathbf{X},\mathbf{Y};\mathbf{Z}) \\ \text{subject to} & \mathbf{X} - \mathbf{Z} - \mathbf{Y} \\ \text{variables} & q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \end{array}}$$

1. Replace $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$ with the model consistency

Derivation

- For each model $p_{\text{model}} \in \{p_{\to\text{xy}}, p_{\text{x}\to\text{y}}, p_{\text{y}\to\text{x}}\}$:

minimize	$I_{\text{xy}\to}(\mathbf{X},\mathbf{Y};\mathbf{Z})$
subject to	$p_{\text{model}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \equiv q_{\text{xy}\to}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v})$
variables	$q_\phi(\mathbf{z}\mid\mathbf{x},\mathbf{y}), q_\phi(\mathbf{u}\mid\mathbf{z},\mathbf{x}), q_\phi(\mathbf{v}\mid\mathbf{z},\mathbf{y}), p_{\text{model}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v})$

1. Replace $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$ with the model consistency

Derivation

- For each model $p_{\mathsf{model}} \in \{p_{\to \mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$:

$$\boxed{\begin{array}{ll} \text{minimize} & I_{\mathsf{xy}\to}(\mathbf{X},\mathbf{Y};\mathbf{Z}) \\ \text{subject to} & D(p_{\mathsf{model}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}), q_{\mathsf{xy}\to}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v})) = 0 \\ \text{variables} & q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}), q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x}), q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y}), p_{\mathsf{model}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \end{array}}$$

1. Replace $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$ with the model consistency

Derivation

- For each model $p_{\text{model}} \in \{p_{\rightarrow\text{xy}}, p_{\text{x}\rightarrow\text{y}}, p_{\text{y}\rightarrow\text{x}}\}$:

$$\boxed{\begin{array}{ll} \text{minimize} & I_{\text{xy}\rightarrow}(\mathbf{X},\mathbf{Y};\mathbf{Z}) \\ \text{subject to} & D(p_{\text{model}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}), q_{\text{xy}\rightarrow}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v})) = 0 \\ \text{variables} & q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}), q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x}), q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y}), p_{\text{model}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \end{array}}$$

1. Replace $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$ with the model consistency
2. Replace $I_{\text{xy}\rightarrow}(\mathbf{X},\mathbf{Y};\mathbf{Z})$ with $I_{\text{model}}(\mathbf{X},\mathbf{Y};\mathbf{Z})$

Derivation

- For each model $p_{\mathsf{model}} \in \{p_{\to\mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$:

minimize	$I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$
subject to	$D(p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}), q_{\mathsf{xy}\to}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})) = 0$
variables	$q_\phi(\mathbf{z}\|\mathbf{x}, \mathbf{y}), q_\phi(\mathbf{u}\|\mathbf{z}, \mathbf{x}), q_\phi(\mathbf{v}\|\mathbf{z}, \mathbf{y}), p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})$

1. Replace $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$ with the model consistency
2. Replace $I_{\mathsf{xy}\to}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ with $I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$

Derivation

- For each model $p_{\mathsf{model}} \in \{p_{\to\mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$:

$$\boxed{\begin{array}{ll} \text{minimize} & I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z}) \\ \text{subject to} & D(p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}), q_{\mathsf{xy}\to}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})) = 0 \\ \text{variables} & q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}), q_\phi(\mathbf{u}|\mathbf{z}, \mathbf{x}), q_\phi(\mathbf{v}|\mathbf{z}, \mathbf{y}), p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}) \end{array}}$$

1. Replace $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$ with the model consistency
2. Replace $I_{\mathsf{xy}\to}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ with $I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$
3. Relax the equality constraint

Derivation

- For each model $p_{\text{model}} \in \{p_{\to\text{xy}}, p_{\text{x}\to\text{y}}, p_{\text{y}\to\text{x}}\}$:

$$
\boxed{\begin{array}{ll}
\text{minimize} & I_{\text{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z}) \\
\text{subject to} & D(p_{\text{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}), q_{\text{xy}\to}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})) \le \epsilon \\
\text{variables} & q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}), q_\phi(\mathbf{u}|\mathbf{z}, \mathbf{x}), q_\phi(\mathbf{v}|\mathbf{z}, \mathbf{y}), p_{\text{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})
\end{array}}
$$

1. Replace $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$ with the model consistency
2. Replace $I_{\text{xy}\to}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ with $I_{\text{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$
3. Relax the equality constraint

Derivation

- For each model $p_{\mathsf{model}} \in \{p_{\to\mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$:

$$\boxed{\begin{array}{ll} \text{minimize} & I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z}) \\ \text{subject to} & D(p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}), q_{\mathsf{xy}\to}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})) \le \epsilon \\ \text{variables} & q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}), q_\phi(\mathbf{u}|\mathbf{z}, \mathbf{x}), q_\phi(\mathbf{v}|\mathbf{z}, \mathbf{y}), p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}) \end{array}}$$

1. Replace $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$ with the model consistency
2. Replace $I_{\mathsf{xy}\to}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ with $I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$
3. Relax the equality constraint
4. Convert to an unconstrained Lagrangian minimization

Derivation

- For each model $p_{\mathsf{model}} \in \{p_{\to\mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$:

$$\boxed{\begin{array}{ll} \text{minimize} & D(p_{\mathsf{model}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}), q_{\mathsf{xy}\to}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v})) + \lambda_{\mathsf{model}}^{\mathsf{CI}} I_{\mathsf{model}}(\mathbf{X},\mathbf{Y};\mathbf{Z}) \\ \text{subject to} & \\ \text{variables} & q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}), q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x}), q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y}), p_{\mathsf{model}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) \end{array}}$$

1. Replace $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$ with the model consistency
2. Replace $I_{\mathsf{xy}\to}(\mathbf{X},\mathbf{Y};\mathbf{Z})$ with $I_{\mathsf{model}}(\mathbf{X},\mathbf{Y};\mathbf{Z})$
3. Relax the equality constraint
4. Convert to an unconstrained Lagrangian minimization

Experiment. CUB image-caption

- $(\mathbf{X}, \mathbf{Y})$=(bird images, captions)

the bird has a white body, black wings, and webbed orange feet

a blue bird with gray primaries and secondaries and white breast and throat

- Used ResNet-101 features for images

Experiment. CUB image-caption

→(image, caption)

this small bird is black white white with a small bill bill and black feet	this bird is grey with grey and a black beak , pointy short pointy beak .	this is a black and white black bird and a short black beak .	this bird has a black and and white and white feathers and

this white bird is mostly white white with a long bill , and black feet	this bird is grey with grey and has long long, pointy short pointy beak	this is a black and white black bird and a long long yellow . .	this bird has a white and and white and white with and feet .

image→caption

input image from test set	generated captions		
	this bird has a black crown and breast , with a crown , and and and black red its . .	this is a very , and white and and color with with a , and and a long blue patches . .	this bird has a very , thin beak with a breast and a brown beak , the body rimmed body .
	this bird has a black crown and breast , with yellow breast and and and its of its feathers .	this bird a small , and yellow black color with with crown black black and black of its crown .	this bird has yellow small , black beak and a breast and a black feathers . the bird it's the body . .
	this bird has a red crown and breast , with red red red and red and on on red its . .	this bird is a red red , red red color with with crown red and and and black red red .	the bird has red red red red , and red red and a red beak . the red 's feathers .

caption→image

input text from test set	ground truth / retrievals from generated features
This bird has yellow topped black and white striped wings and some red markings on its belly.	
This bird has wings that are gray and has a white belly.	

Experiment. CUB image-caption

- Numerical evaluation: correlation of generated samples

Model	joint	image→caption	caption→image
Test set		0.273	
MMVAE [Shi+19]	0.263	0.104	0.135
Variational Wyner	**0.303**	**0.327**	**0.318**