Smoothing Probability Distributions for High Dimensional Learning and Inference

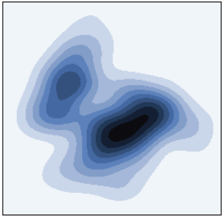
Ziv Goldfeld

Cornell University

CS Brown Bag Talk

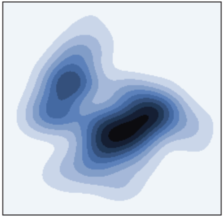
December 1st, 2020

Smoothing Probability Distributions



Data Distribution: $P \in \mathcal{P}(\mathbb{R}^d)$ where $d \gg 1$

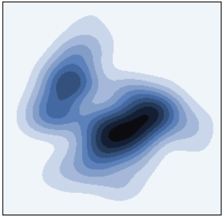
Smoothing Probability Distributions



Data Distribution: $P \in \mathcal{P}(\mathbb{R}^d)$ where $d \gg 1$

'Learning' Objective: Loss, info. measure, distance...

Smoothing Probability Distributions

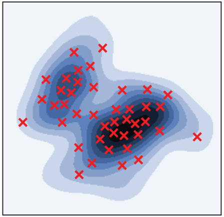


Data Distribution: $P \in \mathcal{P}(\mathbb{R}^d)$ where $d \gg 1$

'Learning' Objective: Loss, info. measure, distance...

Estimation: We don't have P but i.i.d. data $\{X_i\}_{i=1}^n$

Smoothing Probability Distributions

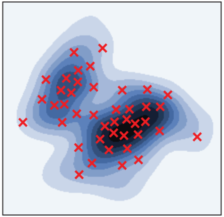


Data Distribution: $P \in \mathcal{P}(\mathbb{R}^d)$ where $d \gg 1$

'Learning' Objective: Loss, info. measure, distance...

Estimation: We don't have P but i.i.d. data $\{X_i\}_{i=1}^n$

Smoothing Probability Distributions



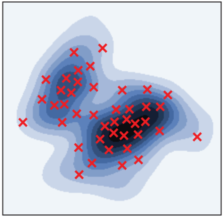
Data Distribution: $P \in \mathcal{P}(\mathbb{R}^d)$ where $d \gg 1$

'Learning' Objective: Loss, info. measure, distance...

Estimation: We don't have P but i.i.d. data $\{X_i\}_{i=1}^n$

$\Longrightarrow$ Estimate objective based on $P_n := \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}$

Smoothing Probability Distributions



Data Distribution: $P \in \mathcal{P}(\mathbb{R}^d)$ where $d \gg 1$

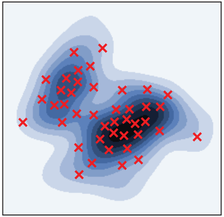
'Learning' Objective: Loss, info. measure, distance...

Estimation: We don't have P but i.i.d. data $\{X_i\}_{i=1}^n$

$\Longrightarrow$ Estimate objective based on $P_n := \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}$

⊛ **Estimation error is typically $n^{-1/d}$**

Smoothing Probability Distributions



Data Distribution: $P \in \mathcal{P}(\mathbb{R}^d)$ where $d \gg 1$

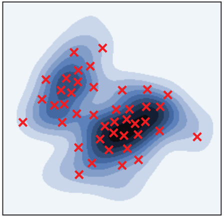
'Learning' Objective: Loss, info. measure, distance...

Estimation: We don't have P but i.i.d. data $\{X_i\}_{i=1}^n$

$\Longrightarrow$ Estimate objective based on $P_n := \frac{1}{n}\sum_{i=1}^{n}\delta_{X_i}$

⊛ **Estimation error is typically $n^{-1/d}$**

Smoothing Probability Distributions



Data Distribution: $P \in \mathcal{P}(\mathbb{R}^d)$ where $d \gg 1$

'Learning' Objective: Loss, info. measure, distance...

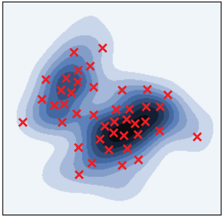
Estimation: We don't have P but i.i.d. data $\{X_i\}_{i=1}^n$

$\Longrightarrow$ Estimate objective based on $P_n := \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}$

⊛ **Estimation error is typically $n^{-1/d}$**

Smoothing: Use $P * \mathcal{N}_\sigma$ and $P_n * \mathcal{N}_\sigma$, $\mathcal{N}_\sigma = \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ ($X+Z$ replaces X)

Smoothing Probability Distributions



Data Distribution: $P \in \mathcal{P}(\mathbb{R}^d)$ where $d \gg 1$

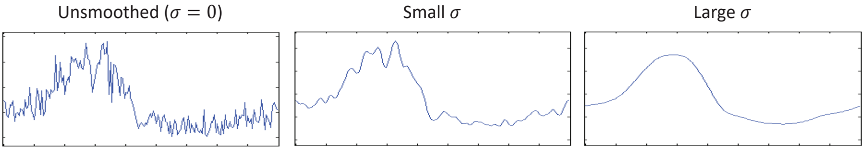
'Learning' Objective: Loss, info. measure, distance...

Estimation: We don't have P but i.i.d. data $\{X_i\}_{i=1}^n$

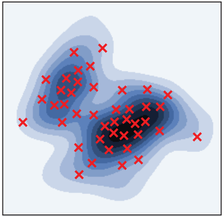
$\Longrightarrow$ Estimate objective based on $P_n := \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}$

⊛ **Estimation error is typically $n^{-1/d}$**

Smoothing: Use $P*\mathcal{N}_\sigma$ and $P_n*\mathcal{N}_\sigma$, $\mathcal{N}_\sigma = \mathcal{N}(0,\sigma^2 \mathrm{I}_d)$ ($X+Z$ replaces X)



Smoothing Probability Distributions



Data Distribution: $P \in \mathcal{P}(\mathbb{R}^d)$ where $d \gg 1$

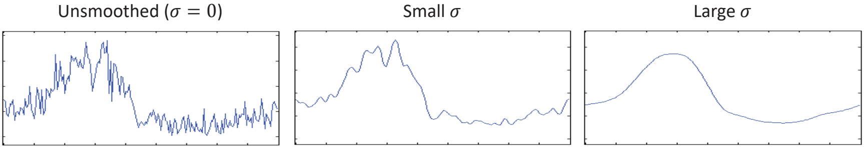
'Learning' Objective: Loss, info. measure, distance...

Estimation: We don't have P but i.i.d. data $\{X_i\}_{i=1}^n$

$\Longrightarrow$ Estimate objective based on $P_n := \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}$

⊛ **Estimation error is typically $n^{-1/d}$**

Smoothing: Use $P*\mathcal{N}_\sigma$ and $P_n*\mathcal{N}_\sigma$, $\mathcal{N}_\sigma = \mathcal{N}(0,\sigma^2\mathrm{I}_d)$ ($X+Z$ replaces X)



Alleviates CoD: Enhancing empirical convergence to $n^{-1/2}$ $\forall d$

Part I:

Measuring Information Flows in Smoothed Deep Neural Networks

Deep Learning - What's Under the Hood?

- Unprecedented practical success

Deep Learning - What's Under the Hood?

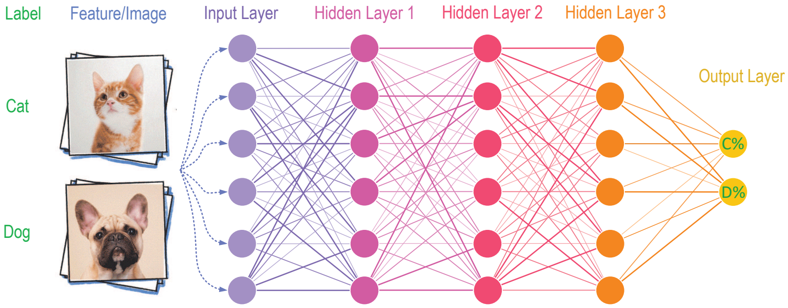
- Unprecedented practical success

Deep Learning - What's Under the Hood?

- Unprecedented practical success
- **Lacking Theory:** Macroscopic understanding of deep learning

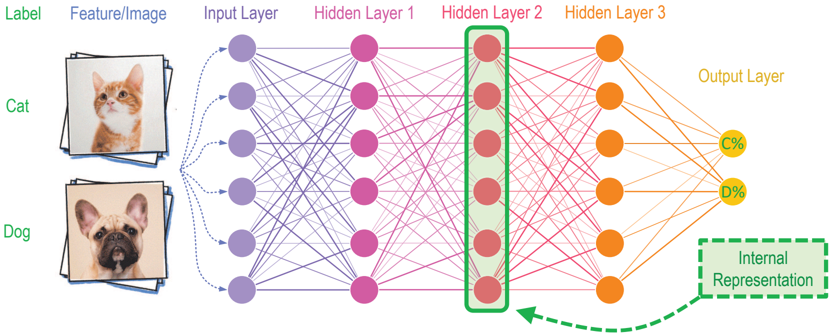
Deep Learning - What's Under the Hood?

- Unprecedented practical success
- **Lacking Theory:** Macroscopic understanding of deep learning



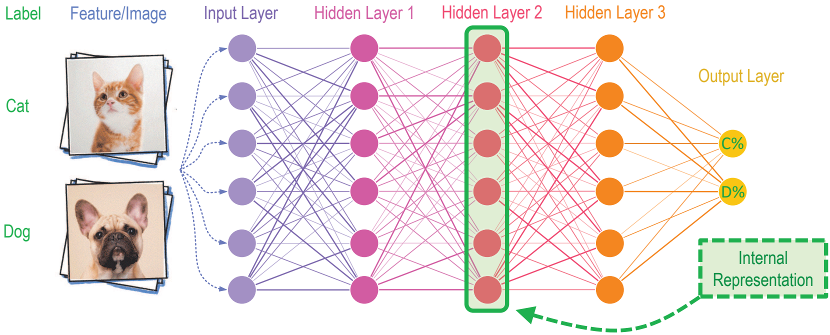
Deep Learning - What's Under the Hood?

- Unprecedented practical success
- **Lacking Theory:** Macroscopic understanding of deep learning



Deep Learning - What's Under the Hood?

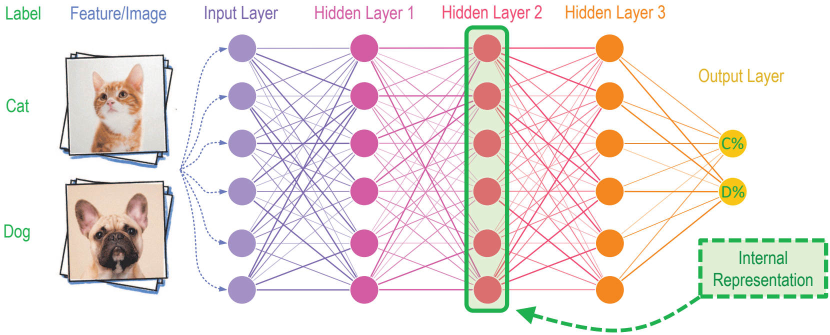
- Unprecedented practical success
- **Lacking Theory:** Macroscopic understanding of deep learning



? What drives the evolution of internal representations?

Deep Learning - What's Under the Hood?

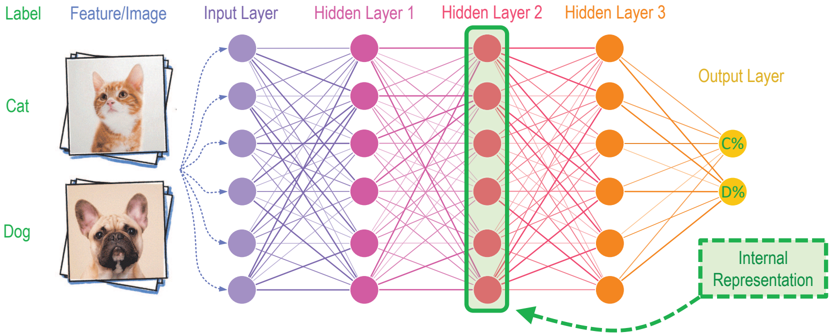
- Unprecedented practical success
- **Lacking Theory:** Macroscopic understanding of deep learning



- ❓ What drives the evolution of internal representations?
- ❓ What are properties of learned representations?

Deep Learning - What's Under the Hood?

- Unprecedented practical success
- **Lacking Theory:** Macroscopic understanding of deep learning



- What drives the evolution of internal representations?
- What are properties of learned representations?
- How fully trained networks process information?

Deep Learning - What's Under the Hood? Cntd.

Trying to Understand Effectiveness of DL:

Deep Learning - What's Under the Hood? Cntd.

Trying to Understand Effectiveness of DL:

- **Statistical learning theory:** Over-parametrization and double descent
  [Belkin-Hsu-Ma'18, Liang-Rakhlin'18, Bartlett-Long-Lugosi-Tsiglera'20]

Deep Learning - What's Under the Hood? Cntd.

Trying to Understand Effectiveness of DL:

- **Statistical learning theory:** Over-parametrization and double descent

  [Belkin-Hsu-Ma'18, Liang-Rakhlin'18, Bartlett-Long-Lugosi-Tsiglera'20]

- **Optimization theory:** Dynamics in parameter space

  [Saxe-McClelland-Ganguli'14, Foster-Sekhari-Sridharan'18, Li-Liang'18]

Deep Learning - What's Under the Hood? Cntd.

Trying to Understand Effectiveness of DL:

- **Statistical learning theory:** Over-parametrization and double descent
  [Belkin-Hsu-Ma'18, Liang-Rakhlin'18, Bartlett-Long-Lugosi-Tsiglera'20]
- **Optimization theory:** Dynamics in parameter space
  [Saxe-McClelland-Ganguli'14, Foster-Sekhari-Sridharan'18, Li-Liang'18]
- **Approximation theory:** Efficiently representable functions
  [Hajnal-et al'93, Delalleau-Bengio'11, Eldan-Shamir'15, Telgarsky'16, Poggio-et al'17]

Deep Learning - What's Under the Hood? Cntd.

Trying to Understand Effectiveness of DL:

- **Statistical learning theory:** Over-parametrization and double descent
  [Belkin-Hsu-Ma'18, Liang-Rakhlin'18, Bartlett-Long-Lugosi-Tsiglera'20]
- **Optimization theory:** Dynamics in parameter space
  [Saxe-McClelland-Ganguli'14, Foster-Sekhari-Sridharan'18, Li-Liang'18]
- **Approximation theory:** Efficiently representable functions
  [Hajnal-et al'93, Delalleau-Bengio'11, Eldan-Shamir'15, Telgarsky'16, Poggio-et al'17]
- **Information theory:** Track information flows through the network
  [Tishby-Zaslavsky'15, Shwartz-Tishby'17, Saxe *et al.*'18, Goldfeld *et al.*'19]

Deep Learning - What's Under the Hood? Cntd.

Trying to Understand Effectiveness of DL:

- **Statistical learning theory:** Over-parametrization and double descent
  [Belkin-Hsu-Ma'18, Liang-Rakhlin'18, Bartlett-Long-Lugosi-Tsiglera'20]
- **Optimization theory:** Dynamics in parameter space
  [Saxe-McClelland-Ganguli'14, Foster-Sekhari-Sridharan'18, Li-Liang'18]
- **Approximation theory:** Efficiently representable functions
  [Hajnal-et al'93, Delalleau-Bengio'11, Eldan-Shamir'15, Telgarsky'16, Poggio-et al'17]
- **Information theory:** Track information flows through the network
  [Tishby-Zaslavsky'15, Shwartz-Tishby'17, Saxe *et al.*'18, Goldfeld *et al.*'19]
  - Information-theoretic complexity measures of representations

Deep Learning - What's Under the Hood? Cntd.

Trying to Understand Effectiveness of DL:

- **Statistical learning theory:** Over-parametrization and double descent
  [Belkin-Hsu-Ma'18, Liang-Rakhlin'18, Bartlett-Long-Lugosi-Tsiglera'20]
- **Optimization theory:** Dynamics in parameter space
  [Saxe-McClelland-Ganguli'14, Foster-Sekhari-Sridharan'18, Li-Liang'18]
- **Approximation theory:** Efficiently representable functions
  [Hajnal-et al'93, Delalleau-Bengio'11, Eldan-Shamir'15, Telgarsky'16, Poggio-et al'17]
- **Information theory:** Track information flows through the network
  [Tishby-Zaslavsky'15, Shwartz-Tishby'17, Saxe *et al.*'18, Goldfeld *et al.*'19]
  - Information-theoretic complexity measures of representations
  - New generalization bounds, architectures, and algorithms

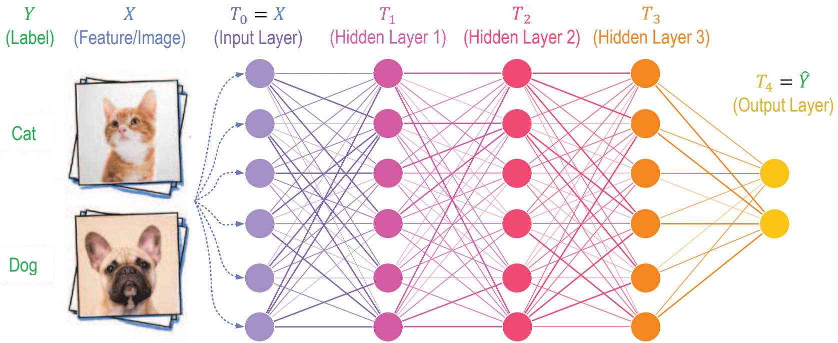
Deep Learning - What's Under the Hood? Cntd.

Trying to Understand Effectiveness of DL:

- **Statistical learning theory:** Over-parametrization and double descent
  [Belkin-Hsu-Ma'18, Liang-Rakhlin'18, Bartlett-Long-Lugosi-Tsiglera'20]
- **Optimization theory:** Dynamics in parameter space
  [Saxe-McClelland-Ganguli'14, Foster-Sekhari-Sridharan'18, Li-Liang'18]
- **Approximation theory:** Efficiently representable functions
  [Hajnal-et al'93, Delalleau-Bengio'11, Eldan-Shamir'15, Telgarsky'16, Poggio-et al'17]
- **Information theory:** Track information flows through the network
  [Tishby-Zaslavsky'15, Shwartz-Tishby'17, Saxe *et al.*'18, Goldfeld *et al.*'19]
  - Information-theoretic complexity measures of representations
  - New generalization bounds, architectures, and algorithms
  - Visualization and interpertability

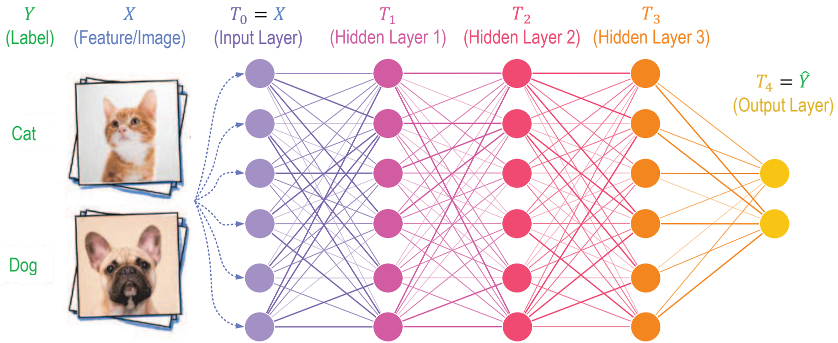
Information Flows in DNNs: Definition

(Deterministic) Feedforward DNN: Each layer $T_\ell = f_\ell(T_{\ell-1})$



Information Flows in DNNs: Definition

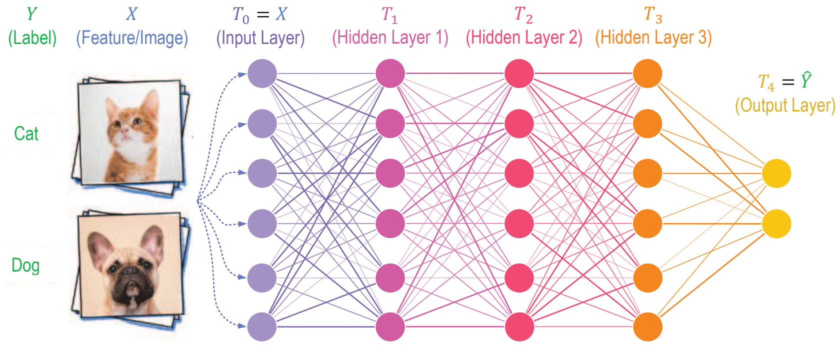
(Deterministic) Feedforward DNN: Each layer $T_\ell = f_\ell(T_{\ell-1})$



- **Joint Distribution:** $P_{X,Y}$

Information Flows in DNNs: Definition

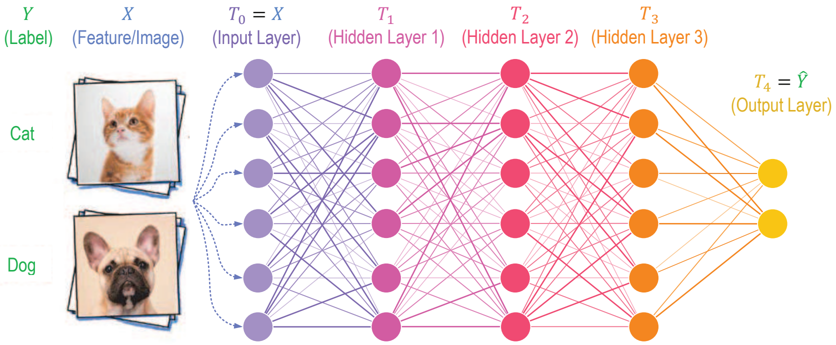
(Deterministic) Feedforward DNN: Each layer $T_\ell = f_\ell(T_{\ell-1})$



- **Joint Distribution:** $P_{X,Y} \implies P_{X,Y} \cdot P_{T_1,\ldots,T_L|X}$

Information Flows in DNNs: Definition

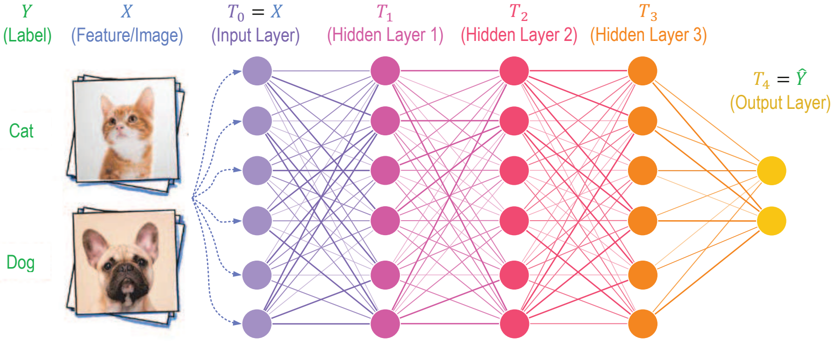
(Deterministic) Feedforward DNN: Each layer $T_\ell = f_\ell(T_{\ell-1})$



- **Joint Distribution:** $P_{X,Y} \implies P_{X,Y} \cdot P_{T_1,\ldots,T_L|X}$
- **Information Flows:** $I(X;T_\ell)$, $I(Y;T_\ell)$, and $I(T_k;T_\ell)$.

Information Flows in DNNs: Definition

<u>(Deterministic) Feedforward DNN:</u> Each layer $T_\ell = f_\ell(T_{\ell-1})$

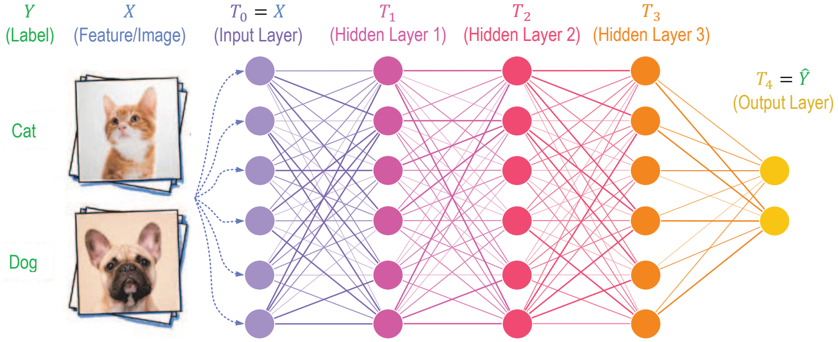


- **Joint Distribution:** $P_{X,Y} \implies P_{X,Y} \cdot P_{T_1,\ldots,T_L|X}$
- **Information Flows:** $I(X;T_\ell)$, $I(Y;T_\ell)$, and $I(T_k;T_\ell)$.

$$\left[I(A;B) = \mathsf{D}_{\mathsf{KL}}\left(P_{A,B} \| P_A \otimes P_B\right) \overset{\text{Discrete}}{=} \sum_{a,b} P_{A,B}(a,b) \log \frac{P_{A,B}(a,b)}{P_A(a)P_B(b)} \right]$$

Information Flows in DNNs: Definition

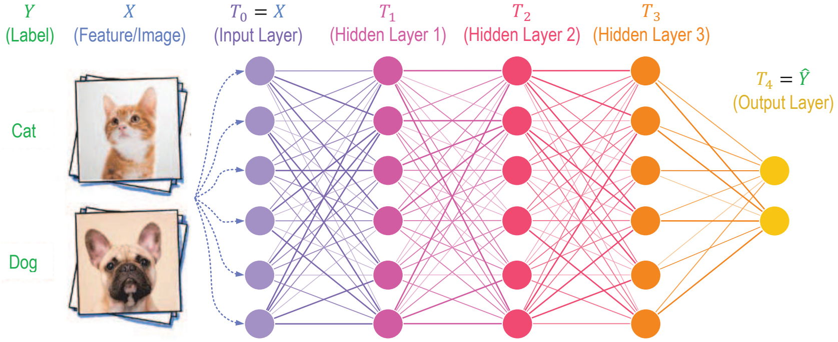
(Deterministic) Feedforward DNN: Each layer $T_\ell = f_\ell(T_{\ell-1})$



- **Joint Distribution:** $P_{X,Y} \implies P_{X,Y} \cdot P_{T_1,\ldots,T_L|X}$
- **Information Flows:** $\boldsymbol{I(X;T_\ell)}$, $\boldsymbol{I(Y;T_\ell)}$, and $I(T_k;T_\ell)$.

Information Flows in DNNs: Definition

(Deterministic) Feedforward DNN: Each layer $T_\ell = f_\ell(T_{\ell-1})$

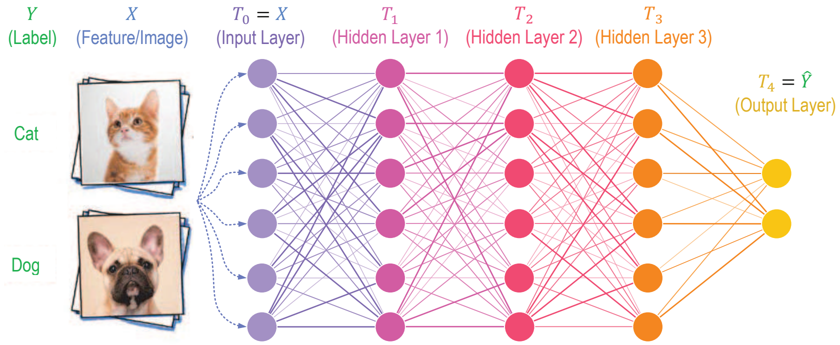


- **Joint Distribution:** $P_{X,Y} \quad \Longrightarrow \quad P_{X,Y} \cdot P_{T_1,\ldots,T_L|X}$
- **Information Flows:** $\boldsymbol{I(X;T_\ell)}$, $\boldsymbol{I(Y;T_\ell)}$, and $I(T_k;T_\ell)$.

Data Processing Inequality: $\boldsymbol{I(Y;T_\ell) \leq I(X;T_\ell)}$

Information Flows in DNNs: Empirical Observations

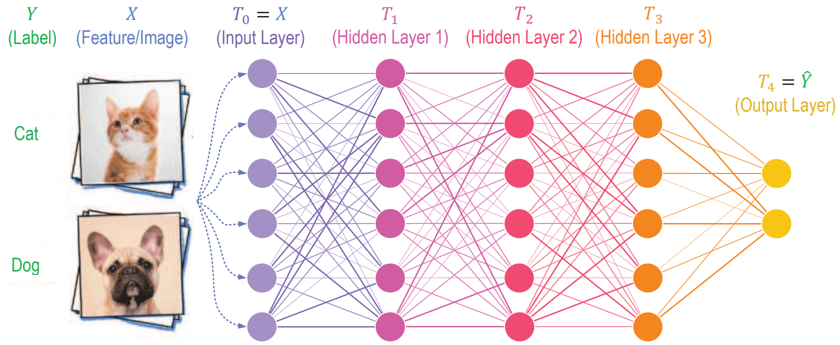
(Deterministic) Feedforward DNN: Each layer $T_\ell = f_\ell(T_{\ell-1})$



Training: Track $\big(I(Y;T_\ell), I(X;T_\ell)\big)$ dynamics

Information Flows in DNNs: Empirical Observations

(Deterministic) Feedforward DNN: Each layer $T_\ell = f_\ell(T_{\ell-1})$

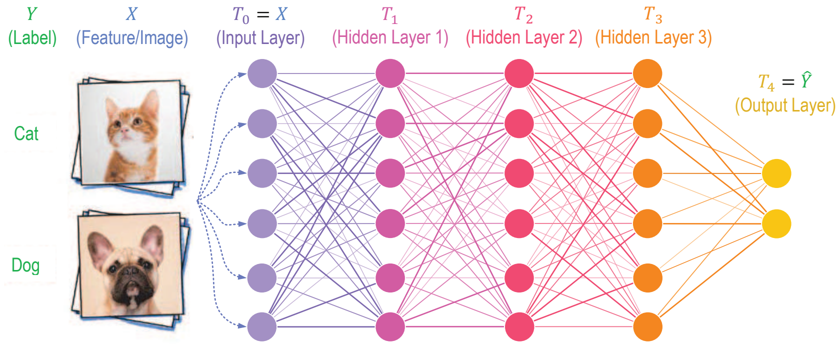


Training: Track $(I(Y;T_\ell), I(X;T_\ell))$ dynamics

1. **Fitting:** $I(Y;T_\ell)$ & $I(X;T_\ell)$ rise (short)

Information Flows in DNNs: Empirical Observations

(Deterministic) Feedforward DNN: Each layer $T_\ell = f_\ell(T_{\ell-1})$

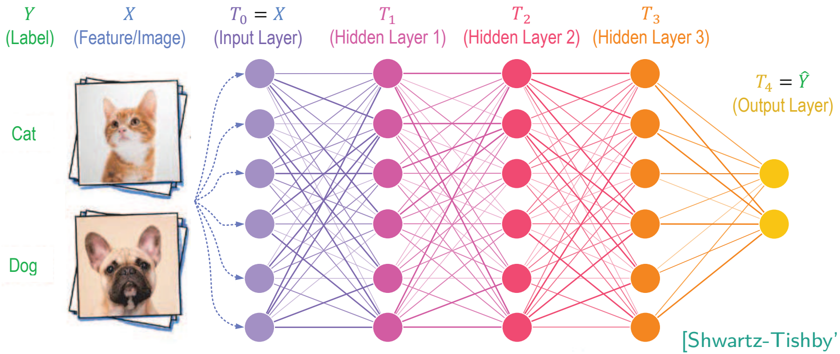


Training: Track $(I(Y;T_\ell), I(X;T_\ell))$ dynamics

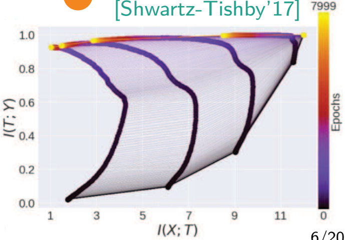
1. **Fitting:** $I(Y;T_\ell)$ & $I(X;T_\ell)$ rise (short)
2. **Compression:** $I(X;T_\ell)$ slowly drops (long)

Information Flows in DNNs: Empirical Observations

<u>**(Deterministic) Feedforward DNN:**</u> Each layer $T_\ell = f_\ell(T_{\ell-1})$



[Shwartz-Tishby'17]



<u>**Training:**</u> Track $(I(Y;T_\ell), I(X;T_\ell))$ dynamics

1. **Fitting:** $I(Y;T_\ell)$ & $I(X;T_\ell)$ rise (short)
2. **Compression:** $I(X;T_\ell)$ slowly drops (long)

Main Challenges and Past Work

Deterministic DNNs: MI degenerates or has $n^{-1/d}$ sample complexity

Main Challenges and Past Work

Deterministic DNNs: MI degenerates or has $n^{-1/d}$ sample complexity

- Past methods are heuristic and w/o accuracy guarantees

Main Challenges and Past Work

Deterministic DNNs: MI degenerates or has $n^{-1/d}$ sample complexity

- Past methods are heuristic and w/o accuracy guarantees

Goal: Meaningful MI **&** Accurate and scalable (in d) estimators

Main Challenges and Past Work

Deterministic DNNs: MI degenerates or has $n^{-1/d}$ sample complexity

- Past methods are heuristic and w/o accuracy guarantees

Goal: Meaningful MI **&** Accurate and scalable (in d) estimators

Smoothing Inject (small) Gaussian noise to neurons' output

[Goldfeld-Berg-Greenewald-Melnyk-Nguyen-Kingsbury-Polyanskiy'19]

Main Challenges and Past Work

Deterministic DNNs: MI degenerates or has $n^{-1/d}$ sample complexity

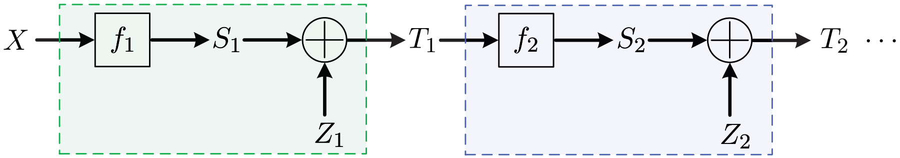
- Past methods are heuristic and w/o accuracy guarantees

Goal: Meaningful MI **&** Accurate and scalable (in d) estimators

Smoothing Inject (small) Gaussian noise to neurons' output

[Goldfeld-Berg-Greenewald-Melnyk-Nguyen-Kingsbury-Polyanskiy'19]

- **Formally:** $T_\ell = S_\ell + Z_\ell$, where $S_\ell := f_\ell(T_{\ell-1})$ and $Z_\ell \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$



Main Challenges and Past Work

Deterministic DNNs: MI degenerates or has $n^{-1/d}$ sample complexity

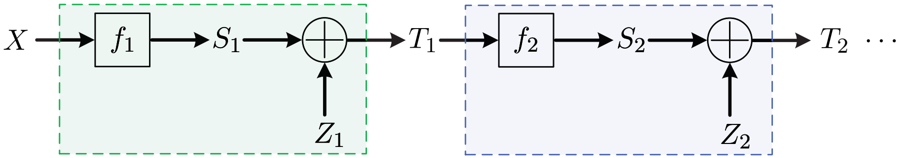
- Past methods are heuristic and w/o accuracy guarantees

Goal: Meaningful MI **&** Accurate and scalable (in d) estimators

Smoothing Inject (small) Gaussian noise to neurons' output

[Goldfeld-Berg-Greenewald-Melnyk-Nguyen-Kingsbury-Polyanskiy'19]

- **Formally:** $T_\ell = S_\ell + Z_\ell$, where $S_\ell := f_\ell(T_{\ell-1})$ and $Z_\ell \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$



$\Longrightarrow$ Good proxy of det. DNN wrt performance & learned representations

Main Challenges and Past Work

Deterministic DNNs: MI degenerates or has $n^{-1/d}$ sample complexity

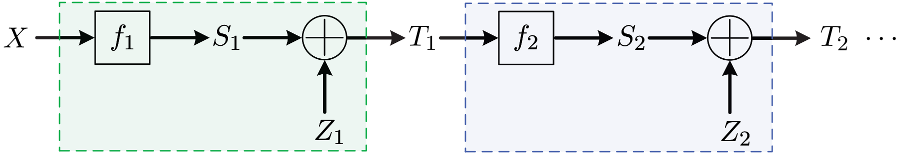
- Past methods are heuristic and w/o accuracy guarantees

Goal: Meaningful MI & Accurate and scalable (in d) estimators

Smoothing Inject (small) Gaussian noise to neurons' output

[Goldfeld-Berg-Greenewald-Melnyk-Nguyen-Kingsbury-Polyanskiy'19]

- **Formally:** $T_\ell = S_\ell + Z_\ell$, where $S_\ell := f_\ell(T_{\ell-1})$ and $Z_\ell \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$



$\Longrightarrow$ Good proxy of det. DNN wrt performance & learned representations

$\Longrightarrow$ Mutual information can be efficiently estimated over noisy DNN!

Mutual Information Estimation - Convergence Rate

Theorem (Goldfeld-Greenewald-Weed-Polyanskiy'20)

For a DNN w/ bdd. activations (tanh/sigmoid), $\sigma > 0$*, and* $\ell = 1, \ldots, L$*:*

$$\inf_{\text{estimator } \hat{I}_\sigma} \sup_{P_X \in \mathcal{P}(\mathbb{R}^d)} \mathbb{E}\left| I(X;T_\ell) - \hat{I}_\sigma(X^n, f_1, \ldots, f_\ell) \right| \leq C_{\sigma,d_\ell} \cdot n^{-\frac{1}{2}}$$

where $X^n := (X_1, \ldots, X_n) \overset{\text{i.i.d.}}{\sim} P_X$ *and* $C_{\sigma,d_\ell} = e^{\Theta(d_\ell)}$*.*

Mutual Information Estimation - Convergence Rate

Theorem (Goldfeld-Greenewald-Weed-Polyanskiy'20)

For a DNN w/ bdd. activations (tanh/sigmoid), $\sigma > 0$*, and* $\ell = 1, \ldots, L$*:*

$$\inf_{\text{estimator } \hat{I}_\sigma} \sup_{P_X \in \mathcal{P}(\mathbb{R}^d)} \mathbb{E}\left| I(X; T_\ell) - \hat{I}_\sigma(X^n, f_1, \ldots, f_\ell) \right| \leq C_{\sigma, d_\ell} \cdot n^{-\frac{1}{2}}$$

where $X^n := (X_1, \ldots, X_n) \overset{\text{i.i.d.}}{\sim} P_X$ *and* $C_{\sigma, d_\ell} = e^{\Theta(d_\ell)}$.

Estimator: Propagate samples & Gaussian conv. w/ empirical measure

Mutual Information Estimation - Convergence Rate

Theorem (Goldfeld-Greenewald-Weed-Polyanskiy'20)

For a DNN w/ bdd. activations (tanh/sigmoid), $\sigma > 0$, and $\ell = 1, \ldots, L$:

$$\inf_{\text{estimator } \hat{I}_\sigma} \sup_{P_X \in \mathcal{P}(\mathbb{R}^d)} \mathbb{E}\left| I(X;T_\ell) - \hat{I}_\sigma(X^n, f_1, \ldots, f_\ell) \right| \leq C_{\sigma,d_\ell} \cdot n^{-\frac{1}{2}}$$

where $X^n := (X_1, \ldots, X_n) \overset{\text{i.i.d.}}{\sim} P_X$ *and* $C_{\sigma,d_\ell} = e^{\Theta(d_\ell)}$.

Estimator: Propagate samples & Gaussian conv. w/ empirical measure

- **Optimal & explicit:** Parametric rate $n^{-1/2}$ & concrete error bounds

Mutual Information Estimation - Convergence Rate

Theorem (Goldfeld-Greenewald-Weed-Polyanskiy'20)

For a DNN w/ bdd. activations (tanh/sigmoid), $\sigma > 0$, and $\ell = 1, \ldots, L$:

$$\inf_{\text{estimator } \hat{I}_\sigma} \sup_{P_X \in \mathcal{P}(\mathbb{R}^d)} \mathbb{E}\left| I(X;T_\ell) - \hat{I}_\sigma(X^n, f_1, \ldots, f_\ell)\right| \leq C_{\sigma,d_\ell} \cdot n^{-\frac{1}{2}}$$

where $X^n := (X_1, \ldots, X_n) \overset{\text{i.i.d.}}{\sim} P_X$ and $C_{\sigma,d_\ell} = e^{\Theta(d_\ell)}$.

Estimator: Propagate samples & Gaussian conv. w/ empirical measure

- **Optimal & explicit:** Parametric rate $n^{-1/2}$ & concrete error bounds
- **Extensions:** Readily adapted for $I(Y;T_\ell)$ and $I(T_k;T_\ell)$ estimation

Mutual Information Estimation - Convergence Rate

Theorem (Goldfeld-Greenewald-Weed-Polyanskiy'20)

For a DNN w/ bdd. activations (tanh/sigmoid), $\sigma > 0$, and $\ell = 1, \ldots, L$:

$$\inf_{\text{estimator } \hat{I}_\sigma} \sup_{P_X \in \mathcal{P}(\mathbb{R}^d)} \mathbb{E}\left|I(X;T_\ell) - \hat{I}_\sigma(X^n, f_1, \ldots, f_\ell)\right| \leq C_{\sigma,d_\ell} \cdot n^{-\frac{1}{2}}$$

where $X^n := (X_1, \ldots, X_n) \overset{i.i.d.}{\sim} P_X$ and $C_{\sigma,d_\ell} = e^{\Theta(d_\ell)}$.

Estimator: Propagate samples & Gaussian conv. w/ empirical measure

- **Optimal & explicit:** Parametric rate $n^{-1/2}$ & concrete error bounds
- **Extensions:** Readily adapted for $I(Y;T_\ell)$ and $I(T_k;T_\ell)$ estimation

Future Goals: Improve scalability in d_ℓ & fast computational algorithm

Mutual Information Estimation - Convergence Rate

Theorem (Goldfeld-Greenewald-Weed-Polyanskiy'20)

For a DNN w/ bdd. activations (tanh/sigmoid), $\sigma > 0$, and $\ell = 1, \ldots, L$:

$$\inf_{\text{estimator } \hat{I}_\sigma} \sup_{P_X \in \mathcal{P}(\mathbb{R}^d)} \mathbb{E}\left| I(X;T_\ell) - \hat{I}_\sigma(X^n, f_1, \ldots, f_\ell) \right| \leq C_{\sigma,d_\ell} \cdot n^{-\frac{1}{2}}$$

where $X^n := (X_1, \ldots, X_n) \overset{\text{i.i.d.}}{\sim} P_X$ and $C_{\sigma,d_\ell} = e^{\Theta(d_\ell)}$.

Estimator: Propagate samples & Gaussian conv. w/ empirical measure

- **Optimal & explicit:** Parametric rate $n^{-1/2}$ & concrete error bounds
- **Extensions:** Readily adapted for $I(Y;T_\ell)$ and $I(T_k;T_\ell)$ estimation

Future Goals: Improve scalability in d_ℓ & fast computational algorithm

⊛ **Scalability:** Manifold hypothesis and/or lower dimensional embeddings

Mutual Information Estimation - Convergence Rate

Theorem (Goldfeld-Greenewald-Weed-Polyanskiy'20)

For a DNN w/ bdd. activations (tanh/sigmoid), $\sigma > 0$, and $\ell = 1, \ldots, L$:

$$\inf_{\text{estimator } \hat{I}_\sigma} \sup_{P_X \in \mathcal{P}(\mathbb{R}^d)} \mathbb{E}\left| I(X;T_\ell) - \hat{I}_\sigma(X^n, f_1, \ldots, f_\ell) \right| \leq C_{\sigma,d_\ell} \cdot n^{-\frac{1}{2}}$$

where $X^n := (X_1, \ldots, X_n) \overset{\text{i.i.d.}}{\sim} P_X$ and $C_{\sigma,d_\ell} = e^{\Theta(d_\ell)}$.

Estimator: Propagate samples & Gaussian conv. w/ empirical measure

- **Optimal & explicit:** Parametric rate $n^{-1/2}$ & concrete error bounds
- **Extensions:** Readily adapted for $I(Y;T_\ell)$ and $I(T_k;T_\ell)$ estimation

Future Goals: Improve scalability in d_ℓ & fast computational algorithm

- **Scalability:** Manifold hypothesis and/or lower dimensional embeddings
- **Algorithms:** Integrate high dimensional Gaussian conv. into DNN arch.

MI Compression vs. Clustering of Representations

Noisy version of DNN from [Shwartz-Tishby'17]:

MI Compression vs. Clustering of Representations

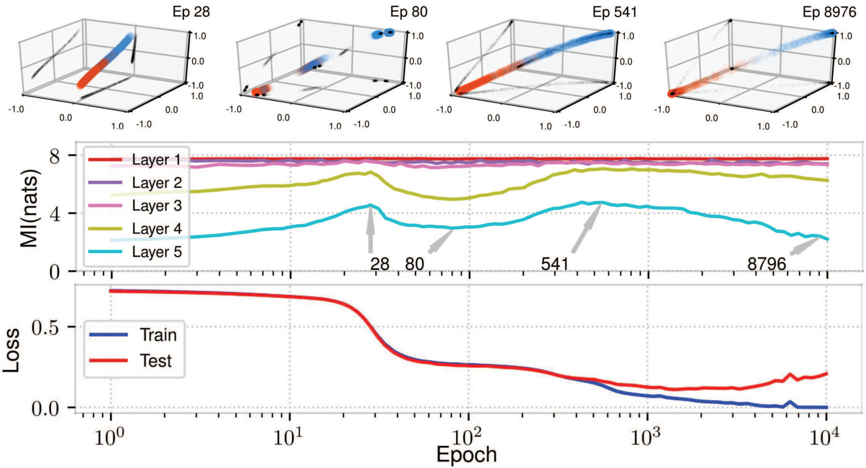
Noisy version of DNN from [Shwartz-Tishby'17]:

- **Binary Classification:** 12-bit input & 12–**10**–**7**–**5**–**4**–**3**–2 tanh MLP

MI Compression vs. Clustering of Representations

Noisy version of DNN from [Shwartz-Tishby'17]:

- **Binary Classification:** 12-bit input & 12–**10–7–5–4–3**–2 tanh MLP



MI Compression vs. Clustering of Representations

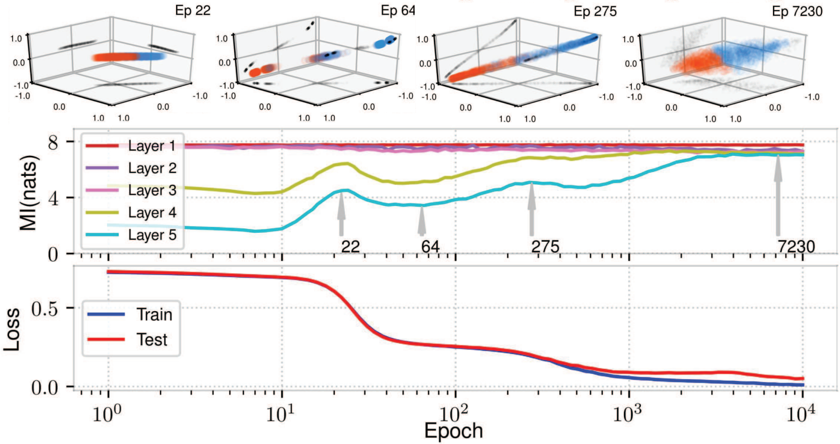
Noisy version of DNN from [Shwartz-Tishby'17]:

- **Binary Classification:** 12-bit input & 12–**10–7–5–4–3**–2 tanh MLP

MI Compression vs. Clustering of Representations

Noisy version of DNN from [Shwartz-Tishby'17]:

- **Binary Classification:** 12-bit input & 12–**10–7–5–4–3**–2 tanh MLP



⊛ weight orthonormality regularization

MI Compression vs. Clustering of Representations

Noisy version of DNN from [Shwartz-Tishby'17]:

- **Binary Classification:** 12-bit input & 12–**10**–**7**–**5**–**4**–**3**–2 tanh MLP
- Verified in multiple experiments

MI Compression vs. Clustering of Representations

Noisy version of DNN from [Shwartz-Tishby'17]:

- **Binary Classification:** 12-bit input & 12–**10**–**7**–**5**–**4**–**3**–2 tanh MLP
- Verified in multiple experiments

$\Longrightarrow$ Compression of $I(X;T_\ell)$ driven by clustering of representations

MI Compression vs. Clustering of Representations

Noisy version of DNN from [Shwartz-Tishby'17]:

- **Binary Classification:** 12-bit input & 12–**10–7–5–4–3**–2 tanh MLP
- Verified in multiple experiments

$\Longrightarrow$ Compression of $I(X;T_\ell)$ driven by clustering of representations

Consequences and Future Goals: $I(X;T_\ell)$ quantifies rep. complexity

MI Compression vs. Clustering of Representations

Noisy version of DNN from [Shwartz-Tishby'17]:

- **Binary Classification:** 12-bit input & 12–**10**–**7**–**5**–**4**–**3**–2 tanh MLP
- Verified in multiple experiments

$\Longrightarrow$ Compression of $I(X;T_\ell)$ driven by clustering of representations

Consequences and Future Goals: $I(X;T_\ell)$ quantifies rep. complexity

⊛ **Prove gen. bounds:** $\mathbb{P}\left(\text{gen}(X^n, Y^n, \mathcal{L}) > \frac{2^{O(I(X;T_\ell))+\delta}}{\sqrt{n}}\right) \lesssim e^{-O(\delta^2)}$

MI Compression vs. Clustering of Representations

Noisy version of DNN from [Shwartz-Tishby'17]:

- **Binary Classification:** 12-bit input & 12–**10**–**7**–**5**–**4**–**3**–2 tanh MLP
- Verified in multiple experiments

$\Longrightarrow$ Compression of $I(X;T_\ell)$ driven by clustering of representations

Consequences and Future Goals: $I(X;T_\ell)$ quantifies rep. complexity

⊛ **Prove gen. bounds:** $\mathbb{P}\left(\text{gen}(X^n, Y^n, \mathcal{L}) > \frac{2^{O(I(X;T_\ell))+\delta}}{\sqrt{n}}\right) \lesssim e^{-O(\delta^2)}$

⊛ **Regularization and prunning:** Algorithmic & architectural advances

MI Compression vs. Clustering of Representations

Noisy version of DNN from [Shwartz-Tishby'17]:

- **Binary Classification:** 12-bit input & 12–**10**–**7**–**5**–**4**–**3**–2 tanh MLP
- Verified in multiple experiments

$\Longrightarrow$ Compression of $I(X;T_\ell)$ driven by clustering of representations

Consequences and Future Goals: $I(X;T_\ell)$ quantifies rep. complexity

⊛ **Prove gen. bounds:** $\mathbb{P}\left(\text{gen}(X^n,Y^n,\mathcal{L}) > \frac{2^{O(I(X;T_\ell))+\delta}}{\sqrt{n}}\right) \lesssim e^{-O(\delta^2)}$

⊛ **Regularization and prunning:** Algorithmic & architectural advances

⊛ **Visualization and interpretability:** Heatmap of DNN neural activity

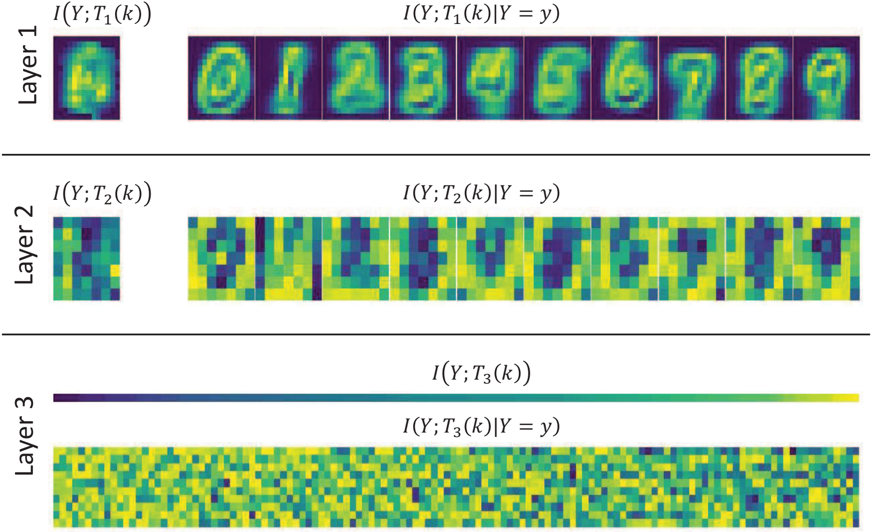
⋮

Mutual Information Heatmap Example

Noisy CNN for MNIST: Classification of hand-written digits

Mutual Information Heatmap Example

Noisy CNN for MNIST: Classification of hand-written digits



Part II:

Smooth Statistical Distances for High-Dimensional Learning and Inference

Implicit (Latent Variable) Generative Models

Goal: Learn a model $Q_\theta \approx P \in \mathcal{P}(\mathbb{R}^d)$ to approximate data distribution

Implicit (Latent Variable) Generative Models

Goal: Learn a model $Q_\theta \approx P \in \mathcal{P}(\mathbb{R}^d)$ to approximate data distribution

Method: Complicated transformation of a simple latent variable

Implicit (Latent Variable) Generative Models

Goal: Learn a model $Q_\theta \approx P \in \mathcal{P}(\mathbb{R}^d)$ to approximate data distribution

Method: Complicated transformation of a simple latent variable

- Latent variable $Z \sim Q_Z \in \mathcal{P}(\mathbb{R}^p)$, $p \ll d$

Implicit (Latent Variable) Generative Models

Goal: Learn a model $Q_\theta \approx P \in \mathcal{P}(\mathbb{R}^d)$ to approximate data distribution

Method: Complicated transformation of a simple latent variable

- Latent variable $Z \sim Q_Z \in \mathcal{P}(\mathbb{R}^p)$, $p \ll d$
- Expand Z to $\mathbb{R}^d$ space via (random) transformation $Q^{(\theta)}_{X|Z}$

Implicit (Latent Variable) Generative Models

Goal: Learn a model $Q_\theta \approx P \in \mathcal{P}(\mathbb{R}^d)$ to approximate data distribution

Method: Complicated transformation of a simple latent variable

- Latent variable $Z \sim Q_Z \in \mathcal{P}(\mathbb{R}^p)$, $p \ll d$
- Expand Z to $\mathbb{R}^d$ space via (random) transformation $Q^{(\theta)}_{X|Z}$

$\Longrightarrow$ **Generative model:** $Q_\theta(\cdot) := \int_{\mathbb{R}^p} Q^{(\theta)}_{X|Z}(\cdot|z)\, \mathrm{d}Q_Z(z)$

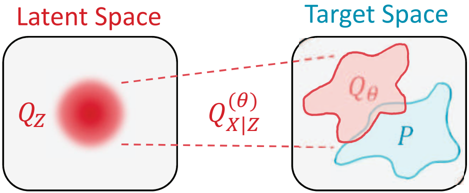
Implicit (Latent Variable) Generative Models

Goal: Learn a model $Q_\theta \approx P \in \mathcal{P}(\mathbb{R}^d)$ to approximate data distribution

Method: Complicated transformation of a simple latent variable

- Latent variable $Z \sim Q_Z \in \mathcal{P}(\mathbb{R}^p)$, $p \ll d$
- Expand Z to $\mathbb{R}^d$ space via (random) transformation $Q^{(\theta)}_{X|Z}$

$\Longrightarrow$ **Generative model:** $Q_\theta(\cdot) := \int_{\mathbb{R}^p} Q^{(\theta)}_{X|Z}(\cdot|z)\, \mathrm{d}Q_Z(z)$



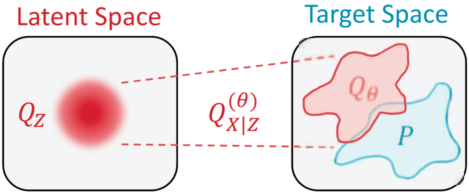
Implicit (Latent Variable) Generative Models

Goal: Learn a model $Q_\theta \approx P \in \mathcal{P}(\mathbb{R}^d)$ to approximate data distribution

Method: Complicated transformation of a simple latent variable

- Latent variable $Z \sim Q_Z \in \mathcal{P}(\mathbb{R}^p)$, $p \ll d$
- Expand Z to $\mathbb{R}^d$ space via (random) transformation $Q^{(\theta)}_{X|Z}$

$\Longrightarrow$ **Generative model:** $Q_\theta(\cdot) := \int_{\mathbb{R}^p} Q^{(\theta)}_{X|Z}(\cdot|z)\, \mathrm{d}Q_Z(z)$



Minimum Distance Estimation: Solve $\theta^\star \in \operatorname*{argmin}_\theta \delta(P, Q_\theta)$

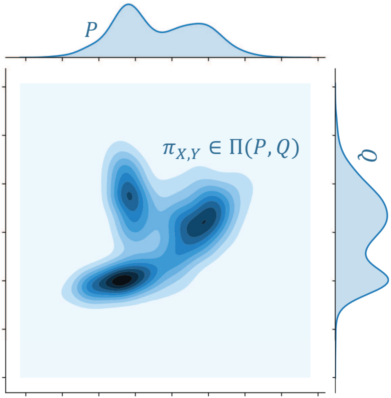
The 1-Wasserstein Distance

Setup: $P, Q \in \mathcal{P}_1(\mathbb{R}^d)$ (subscript for finite 1st moments)

The 1-Wasserstein Distance

Setup: $P, Q \in \mathcal{P}_1(\mathbb{R}^d)$ (subscript for finite 1st moments)

- **Coupling:** $\Pi(P,Q) = \left\{ \pi_{X,Y} \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d) \;\middle|\; \pi_X = P \;\&\; \pi_Y = Q \right\}$



The 1-Wasserstein Distance

Setup: $P, Q \in \mathcal{P}_1(\mathbb{R}^d)$ (subscript for finite 1st moments)

- **Coupling:** $\Pi(P,Q) = \left\{ \pi_{X,Y} \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d) \,\middle|\, \pi_X = P \;\&\; \pi_Y = Q \right\}$
- **Cost:** $c(x,y) = \|x - y\|$ for transporting x to y

The 1-Wasserstein Distance

Setup: $P, Q \in \mathcal{P}_1(\mathbb{R}^d)$ (subscript for finite 1st moments)

- **Coupling:** $\Pi(P, Q) = \left\{\pi_{X,Y} \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d) \;\middle|\; \pi_X = P \;\&\; \pi_Y = Q\right\}$
- **Cost:** $c(x, y) = \|x - y\|$ for transporting x to y

Definition (1-Wasserstein)

The 1-Wasserstein distance: $\mathsf{W}_1(P, Q) := \inf_{\pi_{X,Y} \in \Pi(P,Q)} \mathbb{E}_\pi \|X - Y\|$

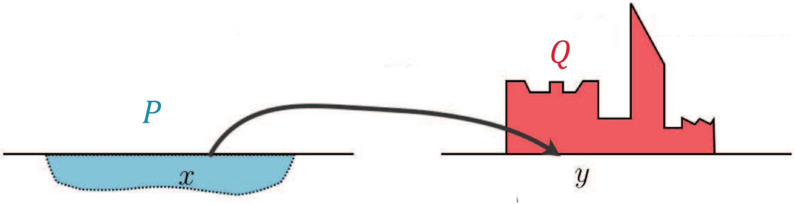
The 1-Wasserstein Distance

Setup: $P, Q \in \mathcal{P}_1(\mathbb{R}^d)$ (subscript for finite 1st moments)

- **Coupling:** $\Pi(P,Q) = \left\{ \pi_{X,Y} \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d) \,\middle|\, \pi_X = P \;\&\; \pi_Y = Q \right\}$
- **Cost:** $c(x,y) = \|x-y\|$ for transporting x to y

Definition (1-Wasserstein)

The 1-Wasserstein distance: $\mathsf{W}_1(P,Q) := \inf_{\pi_{X,Y} \in \Pi(P,Q)} \mathbb{E}_\pi \|X-Y\|$



The 1-Wasserstein Distance

Setup: $P, Q \in \mathcal{P}_1(\mathbb{R}^d)$ (subscript for finite 1st moments)

- **Coupling:** $\Pi(P,Q) = \left\{\pi_{X,Y} \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d) \,\middle|\, \pi_X = P \;\&\; \pi_Y = Q\right\}$
- **Cost:** $c(x,y) = \|x - y\|$ for transporting x to y

Definition (1-Wasserstein)

The 1-Wasserstein distance: $\mathsf{W}_1(P,Q) := \inf_{\pi_{X,Y} \in \Pi(P,Q)} \mathbb{E}_\pi \|X - Y\|$

Comments:

The 1-Wasserstein Distance

Setup: $P, Q \in \mathcal{P}_1(\mathbb{R}^d)$ (subscript for finite 1st moments)

- **Coupling:** $\Pi(P,Q) = \left\{\pi_{X,Y} \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d) \,\middle|\, \pi_X = P \;\&\; \pi_Y = Q\right\}$
- **Cost:** $c(x,y) = \|x-y\|$ for transporting x to y

Definition (1-Wasserstein)

The 1-Wasserstein distance: $\mathsf{W}_1(P,Q) := \inf_{\pi_{X,Y} \in \Pi(P,Q)} \mathbb{E}_\pi \|X-Y\|$

Comments:

- **Robustness to Supp. Mismatch:** $\mathsf{W}_1(P,Q) < \infty$, $\forall P, Q \in \mathcal{P}_1(\mathbb{R}^d)$

The 1-Wasserstein Distance

Setup: $P, Q \in \mathcal{P}_1(\mathbb{R}^d)$ (subscript for finite 1st moments)

- **Coupling:** $\Pi(P,Q) = \left\{\pi_{X,Y} \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d) \,\middle|\, \pi_X = P \;\&\; \pi_Y = Q\right\}$
- **Cost:** $c(x,y) = \|x - y\|$ for transporting x to y

Definition (1-Wasserstein)

The 1-Wasserstein distance: $\mathsf{W}_1(P,Q) := \inf_{\pi_{X,Y} \in \Pi(P,Q)} \mathbb{E}_\pi \|X - Y\|$

Comments:

- **Robustness to Supp. Mismatch:** $\mathsf{W}_1(P,Q) < \infty$, $\forall P, Q \in \mathcal{P}_1(\mathbb{R}^d)$
- **Metric:** $\left(\mathcal{P}_1(\mathbb{R}^d), \mathsf{W}_1\right)$ is metric space (metrizes weak convergence)

The 1-Wasserstein Distance

Setup: $P, Q \in \mathcal{P}_1(\mathbb{R}^d)$ (subscript for finite 1st moments)

- **Coupling:** $\Pi(P,Q) = \left\{\pi_{X,Y} \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d) \,\middle|\, \pi_X = P \;\&\; \pi_Y = Q\right\}$
- **Cost:** $c(x,y) = \|x - y\|$ for transporting x to y

Definition (1-Wasserstein)

The 1-Wasserstein distance: $\mathsf{W}_1(P,Q) := \inf_{\pi_{X,Y} \in \Pi(P,Q)} \mathbb{E}_\pi \|X - Y\|$

Comments:

- **Robustness to Supp. Mismatch:** $\mathsf{W}_1(P,Q) < \infty$, $\forall P, Q \in \mathcal{P}_1(\mathbb{R}^d)$
- **Metric:** $\left(\mathcal{P}_1(\mathbb{R}^d), \mathsf{W}_1\right)$ is metric space (metrizes weak convergence)
- **Duality:** $\mathsf{W}_1(P,Q) = \sup_{f \in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}_P[f] - \mathbb{E}_Q[f] \implies$ **W-GAN** (minimax)

From Duality to Generative Adversarial Networks

Dual Representation: $\mathsf{W}_1(P,Q) = \sup_{f \in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}_P f(X) - \mathbb{E}_Q f(Y)$

From Duality to Generative Adversarial Networks

Dual Representation: $$\mathsf{W}_1(P,Q) = \sup_{f \in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}_P f(X) - \mathbb{E}_Q f(Y)$$

GANs [Goodfellow *et al*'14]:

From Duality to Generative Adversarial Networks

Dual Representation: $\mathsf{W}_1(\boldsymbol{P}, Q) = \sup_{f \in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}_{\boldsymbol{P}} f(\boldsymbol{X}) - \mathbb{E}_Q f(Y)$

GANs [Goodfellow *et al*'14]:

- P (X (real) data sample)

From Duality to Generative Adversarial Networks

Dual Representation: $\mathsf{W}_1(\boldsymbol{P}, \boldsymbol{Q}) = \sup_{f \in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}_{\boldsymbol{P}} f(\boldsymbol{X}) - \mathbb{E}_{\boldsymbol{Q}} f(\boldsymbol{Y})$

GANs [Goodfellow *et al*'14]:

- P (X (real) data sample)
- $Q = Q_\theta$ ($Y = g_\theta(Z)$ gen. sample)

From Duality to Generative Adversarial Networks

Dual Representation: $\mathsf{W}_1(\boldsymbol{P}, \boldsymbol{Q}) = \sup_{\boldsymbol{f} \in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}_{\boldsymbol{P}} \boldsymbol{f}(\boldsymbol{X}) - \mathbb{E}_{\boldsymbol{Q}} \boldsymbol{f}(\boldsymbol{Y})$

GANs [Goodfellow *et al*'14]:

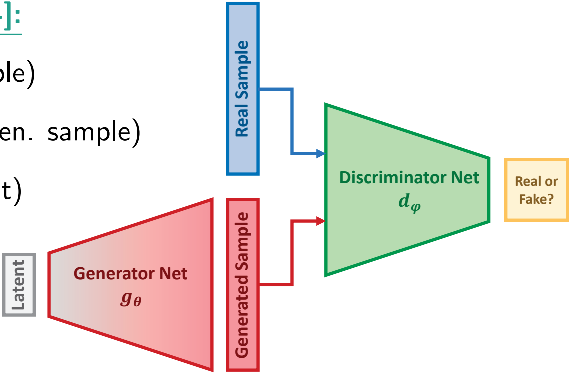
- P (X (real) data sample)
- $Q = Q_\theta$ ($Y = g_\theta(Z)$ gen. sample)
- $f = d_\varphi$ (Lip_1 constraint)

From Duality to Generative Adversarial Networks

Dual Representation: $\mathsf{W}_1(\boldsymbol{P}, \boldsymbol{Q}) = \sup_{\boldsymbol{f} \in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}_{\boldsymbol{P}} \boldsymbol{f}(\boldsymbol{X}) - \mathbb{E}_{\boldsymbol{Q}} \boldsymbol{f}(\boldsymbol{Y})$

GANs [Goodfellow *et al*'14]:

- P (X (real) data sample)
- $Q = Q_\theta$ ($Y = g_\theta(Z)$ gen. sample)
- $f = d_\varphi$ (Lip_1 constraint)

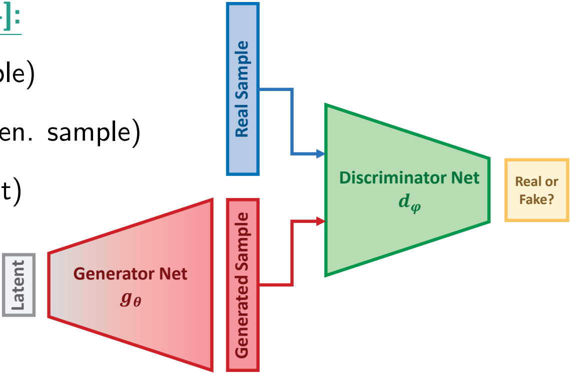


From Duality to Generative Adversarial Networks

Dual Representation: $$\mathsf{W}_1(\boldsymbol{P},\boldsymbol{Q}) = \sup_{\boldsymbol{f}\in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}_{\boldsymbol{P}}\boldsymbol{f}(\boldsymbol{X}) - \mathbb{E}_{\boldsymbol{Q}}\boldsymbol{f}(\boldsymbol{Y})$$

GANs [Goodfellow *et al*'14]:

- P (X (real) data sample)
- $Q = Q_\theta$ ($Y = g_\theta(Z)$ gen. sample)
- $f = d_\varphi$ (Lip_1 constraint)



$$\Longrightarrow \quad \inf_\theta \mathsf{W}_1(P,Q_\theta) \cong \inf_\theta \sup_{\varphi:\ d_\varphi\in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}d_\varphi(X) - \mathbb{E}d_\varphi(g_\theta(Z))$$

Generative Adversarial Networks

NVIDIA's ProGAN 2.0 [Karras *et al*'19]

Implicit Generative Models: Generalization

Goal: Solve $\text{OPT} := \inf_{\theta} \mathsf{W}_1(P, Q_{\theta})$ exactly (find $\theta^{\star}$)

Implicit Generative Models: Generalization

Goal: Solve $\mathsf{OPT} := \inf_\theta \mathsf{W}_1(P, Q_\theta)$ exactly (find $\theta^\star$)

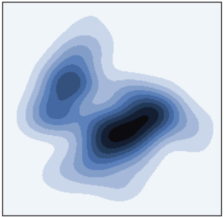
Estimation: We don't have P but data

Implicit Generative Models: Generalization

Goal: Solve OPT $:= \inf_\theta \mathsf{W}_1(P, Q_\theta)$ exactly (find $\theta^\star$)

Estimation: We don't have P but data

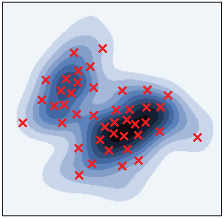
- $\{X_i\}_{i=1}^n$ are i.i.d. samples from $P \in \mathcal{P}(\mathbb{R}^d)$



Implicit Generative Models: Generalization

Goal: Solve $\text{OPT} := \inf_\theta \mathsf{W}_1(P, Q_\theta)$ exactly (find $\theta^\star$)

Estimation: We don't have P but data

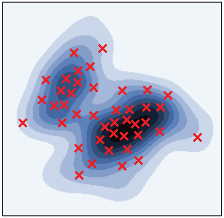


- $\{X_i\}_{i=1}^n$ are i.i.d. samples from $P \in \mathcal{P}(\mathbb{R}^d)$
- Empirical distribution $P_n := \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$

Implicit Generative Models: Generalization

Goal: Solve OPT $:= \inf_\theta \mathsf{W}_1(P, Q_\theta)$ exactly (find $\theta^\star$)

Estimation: We don't have P but data



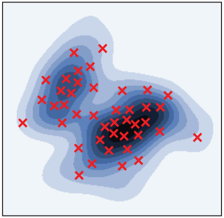
- $\{X_i\}_{i=1}^n$ are i.i.d. samples from $P \in \mathcal{P}(\mathbb{R}^d)$
- Empirical distribution $P_n := \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}$

$\implies$ Inherently we work with $\mathsf{W}_1(P_n, Q_\theta)$

Implicit Generative Models: Generalization

Goal: Solve OPT $:= \inf_\theta \mathsf{W}_1(P, Q_\theta)$ exactly (find $\theta^\star$)

Estimation: We don't have P but data



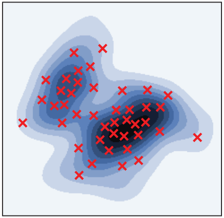
- $\{X_i\}_{i=1}^n$ are i.i.d. samples from $P \in \mathcal{P}(\mathbb{R}^d)$
- Empirical distribution $P_n := \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$

$\implies$ Inherently we work with $\mathsf{W}_1(P_n, Q_\theta)$

Optimization: Can solve $\inf_\theta \mathsf{W}_1(P_n, Q_\theta)$ approximately

Implicit Generative Models: Generalization

Goal: Solve OPT $:= \inf_\theta \mathsf{W}_1(P, Q_\theta)$ exactly (find $\theta^\star$)



Estimation: We don't have P but data

- $\{X_i\}_{i=1}^n$ are i.i.d. samples from $P \in \mathcal{P}(\mathbb{R}^d)$
- Empirical distribution $P_n := \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$

$\implies$ Inherently we work with $\mathsf{W}_1(P_n, Q_\theta)$

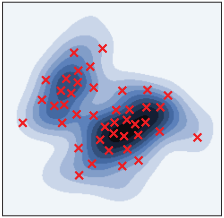
Optimization: Can solve $\inf_\theta \mathsf{W}_1(P_n, Q_\theta)$ approximately

$$\text{Find } \hat{\theta}_n \text{ s.t. } \mathsf{W}_1(P_n, Q_{\hat{\theta}_n}) \le \inf_\theta \mathsf{W}_1(P_n, Q_\theta) + \epsilon$$

Implicit Generative Models: Generalization

Goal: Solve OPT $:= \inf_\theta \mathsf{W}_1(P, Q_\theta)$ exactly (find $\theta^\star$)

Estimation: We don't have P but data



- $\{X_i\}_{i=1}^n$ are i.i.d. samples from $P \in \mathcal{P}(\mathbb{R}^d)$
- Empirical distribution $P_n := \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$

$\implies$ Inherently we work with $\mathsf{W}_1(P_n, Q_\theta)$

Optimization: Can solve $\inf_\theta \mathsf{W}_1(P_n, Q_\theta)$ approximately

$$\text{Find } \hat{\theta}_n \text{ s.t. } \mathsf{W}_1(P_n, Q_{\hat{\theta}_n}) \leq \inf_\theta \mathsf{W}_1(P_n, Q_\theta) + \epsilon$$

Generalization: $\mathsf{W}_1(P, Q_{\hat{\theta}_n}) - \text{OPT} \leq 2\mathsf{W}_1(P_n, P) + \epsilon$

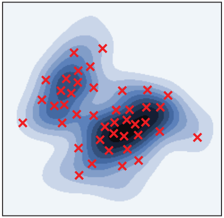
Implicit Generative Models: Generalization

Goal: Solve OPT $:= \inf_\theta \mathsf{W}_1(P, Q_\theta)$ exactly (find $\theta^\star$)

Estimation: We don't have P but data

- $\{X_i\}_{i=1}^n$ are i.i.d. samples from $P \in \mathcal{P}(\mathbb{R}^d)$
- Empirical distribution $P_n := \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$

$\implies$ Inherently we work with $\mathsf{W}_1(P_n, Q_\theta)$



Optimization: Can solve $\inf_\theta \mathsf{W}_1(P_n, Q_\theta)$ approximately

$$\text{Find } \hat{\theta}_n \text{ s.t. } \mathsf{W}_1(P_n, Q_{\hat{\theta}_n}) \leq \inf_\theta \mathsf{W}_1(P_n, Q_\theta) + \epsilon$$

Generalization: $\mathsf{W}_1(P, Q_{\hat{\theta}_n}) - \text{OPT} \leq 2\mathsf{W}_1(P_n, P) + \epsilon$

$\implies$ **Boils down to empirical approximation question under W_1**

Empirical Approximation in High Dimensions

Question: What can we say about $\mathsf{W}_1(P_n, P)$?

Empirical Approximation in High Dimensions

Question: What can we say about $\mathsf{W}_1(P_n, P)$?

Theorem (Dudley'69)

For $d \geq 3$ *and* $\mathcal{P}_1(\mathbb{R}^d) \ni P \ll \mathsf{Leb}(\mathbb{R}^d)$*:* $\mathbb{E}\mathsf{W}_1(P_n, P) \asymp n^{-\frac{1}{d}}$

Empirical Approximation in High Dimensions

Question: What can we say about $\mathsf{W}_1(P_n, P)$?

Theorem (Dudley'69)

For $d \geq 3$ *and* $\mathcal{P}_1(\mathbb{R}^d) \ni P \ll \mathsf{Leb}(\mathbb{R}^d)$*:* $\mathbb{E}\mathsf{W}_1(P_n, P) \asymp n^{-\frac{1}{d}}$

Empirical Approximation in High Dimensions

Question: What can we say about $\mathsf{W}_1(P_n, P)$?

Theorem (Dudley'69)

For $d \geq 3$ *and* $\mathcal{P}_1(\mathbb{R}^d) \ni P \ll \mathsf{Leb}(\mathbb{R}^d)$*:* $\mathbb{E}\mathsf{W}_1(P_n, P) \asymp n^{-\frac{1}{d}}$

⊛ **Implication:** Too slow given dimensionality of real-world data

Empirical Approximation in High Dimensions

Question: What can we say about $\mathsf{W}_1(P_n, P)$?

Theorem (Dudley'69)

For $d \geq 3$ and $\mathcal{P}_1(\mathbb{R}^d) \ni P \ll \mathsf{Leb}(\mathbb{R}^d)$: $\mathbb{E}\mathsf{W}_1(P_n, P) \asymp n^{-\frac{1}{d}}$

⊛ **Implication:** Too slow given dimensionality of real-world data

⊛ **Question:** Can smoothing help alleviates CoD?

Smooth 1-Wasserstein Distance

Definition (Goldfeld-Greenewald'19)

For $\sigma \geq 0$, the smooth 1-Wasserstein distance between P and Q is

$$\mathsf{W}_1^{(\sigma)}(P, Q) := \mathsf{W}_1(P * \mathcal{N}_\sigma, Q * \mathcal{N}_\sigma),$$

where $\mathcal{N}_\sigma := \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ is a d-dimensional isotropic Gaussian.

Smooth 1-Wasserstein Distance

Definition (Goldfeld-Greenewald'19)

For $\sigma \geq 0$, the smooth 1-Wasserstein distance between P and Q is

$$\mathsf{W}_1^{(\sigma)}(P,Q) := \mathsf{W}_1(P * \mathcal{N}_\sigma, Q * \mathcal{N}_\sigma),$$

where $\mathcal{N}_\sigma := \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ is a d-dimensional isotropic Gaussian.

Interpretation: $X \sim P$, $Y \sim Q$ and $Z_1, Z_2 \sim \mathcal{N}_\sigma$

Smooth 1-Wasserstein Distance

Definition (Goldfeld-Greenewald'19)

For $\sigma \geq 0$, the smooth 1-Wasserstein distance between P and Q is

$$\mathsf{W}_1^{(\sigma)}(P, Q) := \mathsf{W}_1(P * \mathcal{N}_\sigma, Q * \mathcal{N}_\sigma),$$

where $\mathcal{N}_\sigma := \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ is a d-dimensional isotropic Gaussian.

Interpretation: $X \sim P$, $Y \sim Q$ and $Z_1, Z_2 \sim \mathcal{N}_\sigma$

$$X \perp Z_1 \implies X + Z_1 \sim P * \mathcal{N}_\sigma \quad \& \quad Y \perp Z_2 \implies Y + Z_2 \sim Q * \mathcal{N}_\sigma$$

Smooth 1-Wasserstein Distance

Definition (Goldfeld-Greenewald'19)

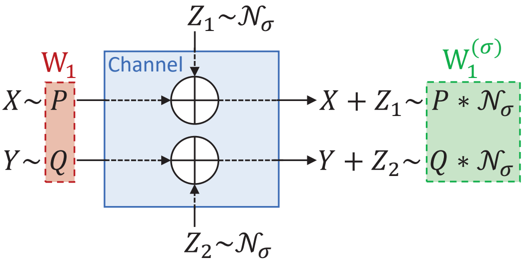
For $\sigma \geq 0$, the smooth 1-Wasserstein distance between P and Q is

$$\mathsf{W}_1^{(\sigma)}(P,Q) := \mathsf{W}_1(P * \mathcal{N}_\sigma, Q * \mathcal{N}_\sigma),$$

where $\mathcal{N}_\sigma := \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ is a d-dimensional isotropic Gaussian.

Interpretation: $X \sim P$, $Y \sim Q$ and $Z_1, Z_2 \sim \mathcal{N}_\sigma$

$$X \perp Z_1 \implies X + Z_1 \sim P * \mathcal{N}_\sigma \quad \& \quad Y \perp Z_2 \implies Y + Z_2 \sim Q * \mathcal{N}_\sigma$$



Smooth 1-Wasserstein Distance

Definition (Goldfeld-Greenewald'19)

For $\sigma \geq 0$, the smooth 1-Wasserstein distance between P and Q is

$$\mathsf{W}_1^{(\sigma)}(P,Q) := \mathsf{W}_1(P * \mathcal{N}_\sigma, Q * \mathcal{N}_\sigma),$$

where $\mathcal{N}_\sigma := \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ is a d-dimensional isotropic Gaussian.

Interpretation: $X \sim P$, $Y \sim Q$ and $Z_1, Z_2 \sim \mathcal{N}_\sigma$

$$X \perp Z_1 \implies X + Z_1 \sim P * \mathcal{N}_\sigma \quad \& \quad Y \perp Z_2 \implies Y + Z_2 \sim Q * \mathcal{N}_\sigma$$

Properties: Preserves structure but enhances statistical convergence

Smooth 1-Wasserstein Distance

Definition (Goldfeld-Greenewald'19)

For $\sigma \geq 0$, the smooth 1-Wasserstein distance between P and Q is

$$\mathsf{W}_1^{(\sigma)}(P,Q) := \mathsf{W}_1(P * \mathcal{N}_\sigma, Q * \mathcal{N}_\sigma),$$

where $\mathcal{N}_\sigma := \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ is a d-dimensional isotropic Gaussian.

Interpretation: $X \sim P$, $Y \sim Q$ and $Z_1, Z_2 \sim \mathcal{N}_\sigma$

$$X \perp Z_1 \implies X + Z_1 \sim P * \mathcal{N}_\sigma \quad \& \quad Y \perp Z_2 \implies Y + Z_2 \sim Q * \mathcal{N}_\sigma$$

Properties: Preserves structure but enhances statistical convergence

- **Retain duality:** $\mathsf{W}_1^{(\sigma)}(P,Q) = \sup_{f \in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}\big[f(X+Z)\big] - \mathbb{E}\big[f(Y+Z)\big]$

Smooth 1-Wasserstein Distance

Definition (Goldfeld-Greenewald'19)

For $\sigma \geq 0$, the smooth 1-Wasserstein distance between P and Q is

$$\mathsf{W}_1^{(\sigma)}(P,Q) := \mathsf{W}_1(P * \mathcal{N}_\sigma, Q * \mathcal{N}_\sigma),$$

where $\mathcal{N}_\sigma := \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ is a d-dimensional isotropic Gaussian.

Interpretation: $X \sim P$, $Y \sim Q$ and $Z_1, Z_2 \sim \mathcal{N}_\sigma$

$$X \perp Z_1 \implies X + Z_1 \sim P * \mathcal{N}_\sigma \quad \& \quad Y \perp Z_2 \implies Y + Z_2 \sim Q * \mathcal{N}_\sigma$$

Properties: Preserves structure but enhances statistical convergence

- **Retain duality:** $\mathsf{W}_1^{(\sigma)}(P,Q) = \sup_{f \in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}[f(X+Z)] - \mathbb{E}[f(Y+Z)]$
- **Inherit metric structure:** Topologically equivalent to unsmooth W_1

Smooth 1-Wasserstein Distance

Definition (Goldfeld-Greenewald'19)

For $\sigma \geq 0$, the smooth 1-Wasserstein distance between P and Q is

$$\mathsf{W}_1^{(\sigma)}(P,Q) := \mathsf{W}_1(P * \mathcal{N}_\sigma, Q * \mathcal{N}_\sigma),$$

where $\mathcal{N}_\sigma := \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ is a d-dimensional isotropic Gaussian.

Interpretation: $X \sim P$, $Y \sim Q$ and $Z_1, Z_2 \sim \mathcal{N}_\sigma$

$$X \perp Z_1 \implies X + Z_1 \sim P * \mathcal{N}_\sigma \quad \& \quad Y \perp Z_2 \implies Y + Z_2 \sim Q * \mathcal{N}_\sigma$$

Properties: Preserves structure but enhances statistical convergence

- **Retain duality:** $\mathsf{W}_1^{(\sigma)}(P,Q) = \sup_{f \in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}\big[f(X+Z)\big] - \mathbb{E}\big[f(Y+Z)\big]$
- **Inherit metric structure:** Topologically equivalent to unsmooth W_1
- **Stability:** $\big|\mathsf{W}_1^{(\sigma)}(P,Q) - \mathsf{W}_1(P,Q)\big| \leq 2\sigma\sqrt{d}$ for all P, Q

Smooth 1-Wasserstein Distance

Definition (Goldfeld-Greenewald'19)

For $\sigma \geq 0$, the smooth 1-Wasserstein distance between P and Q is

$$\mathsf{W}_1^{(\sigma)}(P,Q) := \mathsf{W}_1(P * \mathcal{N}_\sigma, Q * \mathcal{N}_\sigma),$$

where $\mathcal{N}_\sigma := \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ is a d-dimensional isotropic Gaussian.

Interpretation: $X \sim P$, $Y \sim Q$ and $Z_1, Z_2 \sim \mathcal{N}_\sigma$

$$X \perp Z_1 \implies X + Z_1 \sim P * \mathcal{N}_\sigma \quad \& \quad Y \perp Z_2 \implies Y + Z_2 \sim Q * \mathcal{N}_\sigma$$

Properties: Preserves structure but enhances statistical convergence

- **Retain duality:** $\mathsf{W}_1^{(\sigma)}(P,Q) = \sup_{f \in \mathsf{Lip}_1(\mathbb{R}^d)} \mathbb{E}\big[f(X+Z)\big] - \mathbb{E}\big[f(Y+Z)\big]$
- **Inherit metric structure:** Topologically equivalent to unsmooth W_1
- **Stability:** $\big|\mathsf{W}_1^{(\sigma)}(P,Q) - \mathsf{W}_1(P,Q)\big| \leq 2\sigma\sqrt{d}$ for all P, Q
- **Fast emp. convergence:** $\mathsf{W}_1^{(\sigma)}(P_n, P) \asymp n^{-1/2}$ in all dimensions!

Smooth Distances for Generative Modeling

Smooth Generative Models: MDE wrt smooth distance

Smooth Distances for Generative Modeling

Smooth Generative Models: MDE wrt smooth distance

1. **Generalization:** $\mathsf{W}_1^{(\sigma)}(P, Q_{\hat{\theta}_n}) - \inf_\theta \mathsf{W}_1^{(\sigma)}(P, Q_\theta) \lesssim n^{-\frac{1}{2}}, \quad \forall d$

Smooth Distances for Generative Modeling

Smooth Generative Models: MDE wrt smooth distance

1. **Generalization:** $\mathsf{W}_1^{(\sigma)}(P, Q_{\hat{\theta}_n}) - \inf_\theta \mathsf{W}_1^{(\sigma)}(P, Q_\theta) \lesssim n^{-\frac{1}{2}}, \quad \forall d$
2. **Limit distributions:** Asymptotic dist. of MDE and empirical error

Smooth Distances for Generative Modeling

Smooth Generative Models: MDE wrt smooth distance

1. **Generalization:** $\mathsf{W}_1^{(\sigma)}(P, Q_{\hat{\theta}_n}) - \inf_\theta \mathsf{W}_1^{(\sigma)}(P, Q_\theta) \lesssim n^{-\frac{1}{2}}, \quad \forall d$
2. **Limit distributions:** Asymptotic dist. of MDE and empirical error
3. **Inequalities:** Web of relationships between smooth distances

Smooth Distances for Generative Modeling

Smooth Generative Models: MDE wrt smooth distance

1. **Generalization:** $\mathsf{W}_1^{(\sigma)}(P, Q_{\hat{\theta}_n}) - \inf_\theta \mathsf{W}_1^{(\sigma)}(P, Q_\theta) \lesssim n^{-\frac{1}{2}}, \quad \forall d$
2. **Limit distributions:** Asymptotic dist. of MDE and empirical error
3. **Inequalities:** Web of relationships between smooth distances

$\Longrightarrow$ Compatible for high-dimensional learning and inference!

Smooth Distances for Generative Modeling

Smooth Generative Models: MDE wrt smooth distance

1. **Generalization:** $\mathsf{W}_1^{(\sigma)}(P, Q_{\hat{\theta}_n}) - \inf_\theta \mathsf{W}_1^{(\sigma)}(P, Q_\theta) \lesssim n^{-\frac{1}{2}}, \quad \forall d$
2. **Limit distributions:** Asymptotic dist. of MDE and empirical error
3. **Inequalities:** Web of relationships between smooth distances

$\Longrightarrow$ Compatible for high-dimensional learning and inference!

Future Goals: More distances, kernel, and efficient algorithms

Smooth Distances for Generative Modeling

Smooth Generative Models: MDE wrt smooth distance

1. **Generalization:** $\mathsf{W}_1^{(\sigma)}(P, Q_{\hat{\theta}_n}) - \inf_\theta \mathsf{W}_1^{(\sigma)}(P, Q_\theta) \lesssim n^{-\frac{1}{2}}, \quad \forall d$
2. **Limit distributions:** Asymptotic dist. of MDE and empirical error
3. **Inequalities:** Web of relationships between smooth distances

$\Longrightarrow$ Compatible for high-dimensional learning and inference!

Future Goals: More distances, kernel, and efficient algorithms

⊛ **More distances:** p-Wasserstein distances, f-divergences, and IPMs

Smooth Distances for Generative Modeling

Smooth Generative Models: MDE wrt smooth distance

1. **Generalization:** $\mathsf{W}_1^{(\sigma)}(P, Q_{\hat{\theta}_n}) - \inf_\theta \mathsf{W}_1^{(\sigma)}(P, Q_\theta) \lesssim n^{-\frac{1}{2}}, \quad \forall d$
2. **Limit distributions:** Asymptotic dist. of MDE and empirical error
3. **Inequalities:** Web of relationships between smooth distances

$\Longrightarrow$ Compatible for high-dimensional learning and inference!

Future Goals: More distances, kernel, and efficient algorithms

- **More distances:** p-Wasserstein distances, f-divergences, and IPMs
- **More kernels:** Optimize over choice of smoothing kernel

Smooth Distances for Generative Modeling

Smooth Generative Models: MDE wrt smooth distance

1. **Generalization:** $\mathsf{W}_1^{(\sigma)}(P, Q_{\hat{\theta}_n}) - \inf_\theta \mathsf{W}_1^{(\sigma)}(P, Q_\theta) \lesssim n^{-\frac{1}{2}}, \quad \forall d$
2. **Limit distributions:** Asymptotic dist. of MDE and empirical error
3. **Inequalities:** Web of relationships between smooth distances

$\Longrightarrow$ Compatible for high-dimensional learning and inference!

Future Goals: More distances, kernel, and efficient algorithms

- **More distances:** p-Wasserstein distances, f-divergences, and IPMs
- **More kernels:** Optimize over choice of smoothing kernel
- **Efficient algorithms:** Fast computational methods

⋮

Smooth Distances for Generative Modeling

Smooth Generative Models: MDE wrt smooth distance

1. **Generalization:** $\mathsf{W}_1^{(\sigma)}(P, Q_{\hat{\theta}_n}) - \inf_\theta \mathsf{W}_1^{(\sigma)}(P, Q_\theta) \lesssim n^{-\frac{1}{2}}, \quad \forall d$
2. **Limit distributions:** Asymptotic dist. of MDE and empirical error
3. **Inequalities:** Web of relationships between smooth distances

$\Longrightarrow$ Compatible for high-dimensional learning and inference!

Future Goals: More distances, kernel, and efficient algorithms

- **More distances:** p-Wasserstein distances, f-divergences, and IPMs
- **More kernels:** Optimize over choice of smoothing kernel
- **Efficient algorithms:** Fast computational methods

⋮

Next-generation systems: benchmark performance & resource efficiency

Additional Research Topics

Neural Estimation:

Additional Research Topics

Neural Estimation:

- Approx. discriminator by a NN & optimize via gradient methods

Additional Research Topics

Neural Estimation:

- Approx. discriminator by a NN & optimize via gradient methods
- **Performance guarantees?** Approximation vs. estimation tradeoffs

Additional Research Topics

Neural Estimation:

- Approx. discriminator by a NN & optimize via gradient methods
- **Performance guarantees?** Approximation vs. estimation tradeoffs

Learning under privacy:

Additional Research Topics

Neural Estimation:

- Approx. discriminator by a NN & optimize via gradient methods
- **Performance guarantees?** Approximation vs. estimation tradeoffs

Learning under privacy:

- Adapt classic learning setup to incorporate privacy constraints

Additional Research Topics

Neural Estimation:

- Approx. discriminator by a NN & optimize via gradient methods
- **Performance guarantees?** Approximation vs. estimation tradeoffs

Learning under privacy:

- Adapt classic learning setup to incorporate privacy constraints
- **Theory:** Bound the risk when compared to non-privatized learner

Additional Research Topics

Neural Estimation:

- Approx. discriminator by a NN & optimize via gradient methods
- **Performance guarantees?** Approximation vs. estimation tradeoffs

Learning under privacy:

- Adapt classic learning setup to incorporate privacy constraints
- **Theory:** Bound the risk when compared to non-privatized learner
- **Algorithms:** Key-based schemes, Hadamard codes, etc.

Additional Research Topics

Neural Estimation:

- Approx. discriminator by a NN & optimize via gradient methods
- **Performance guarantees?** Approximation vs. estimation tradeoffs

Learning under privacy:

- Adapt classic learning setup to incorporate privacy constraints
- **Theory:** Bound the risk when compared to non-privatized learner
- **Algorithms:** Key-based schemes, Hadamard codes, etc.

Data Storage in Interacting Particle Systems:

Additional Research Topics

Neural Estimation:

- Approx. discriminator by a NN & optimize via gradient methods
- **Performance guarantees?** Approximation vs. estimation tradeoffs

Learning under privacy:

- Adapt classic learning setup to incorporate privacy constraints
- **Theory:** Bound the risk when compared to non-privatized learner
- **Algorithms:** Key-based schemes, Hadamard codes, etc.

Data Storage in Interacting Particle Systems:

- Distill storage question from particular tech. & incorporate physics

Additional Research Topics

Neural Estimation:

- Approx. discriminator by a NN & optimize via gradient methods
- **Performance guarantees?** Approximation vs. estimation tradeoffs

Learning under privacy:

- Adapt classic learning setup to incorporate privacy constraints
- **Theory:** Bound the risk when compared to non-privatized learner
- **Algorithms:** Key-based schemes, Hadamard codes, etc.

Data Storage in Interacting Particle Systems:

- Distill storage question from particular tech. & incorporate physics
- Study **information capacity** (systems size, storage time, temp.)

Additional Research Topics

Neural Estimation:

- Approx. discriminator by a NN & optimize via gradient methods
- **Performance guarantees?** Approximation vs. estimation tradeoffs

Learning under privacy:

- Adapt classic learning setup to incorporate privacy constraints
- **Theory:** Bound the risk when compared to non-privatized learner
- **Algorithms:** Key-based schemes, Hadamard codes, etc.

Data Storage in Interacting Particle Systems:

- Distill storage question from particular tech. & incorporate physics
- Study **information capacity** (systems size, storage time, temp.)

Physical Layer Security:

Additional Research Topics

Neural Estimation:

- Approx. discriminator by a NN & optimize via gradient methods
- **Performance guarantees?** Approximation vs. estimation tradeoffs

Learning under privacy:

- Adapt classic learning setup to incorporate privacy constraints
- **Theory:** Bound the risk when compared to non-privatized learner
- **Algorithms:** Key-based schemes, Hadamard codes, etc.

Data Storage in Interacting Particle Systems:

- Distill storage question from particular tech. & incorporate physics
- Study **information capacity** (systems size, storage time, temp.)

Physical Layer Security:

- Beneficial properties but impractical assumptions (known channel)

Additional Research Topics

Neural Estimation:

- Approx. discriminator by a NN & optimize via gradient methods
- **Performance guarantees?** Approximation vs. estimation tradeoffs

Learning under privacy:

- Adapt classic learning setup to incorporate privacy constraints
- **Theory:** Bound the risk when compared to non-privatized learner
- **Algorithms:** Key-based schemes, Hadamard codes, etc.

Data Storage in Interacting Particle Systems:

- Distill storage question from particular tech. & incorporate physics
- Study **information capacity** (systems size, storage time, temp.)

Physical Layer Security:

- Beneficial properties but impractical assumptions (known channel)
- **Bridge gaps** via adversarial models & connect to adversarial learning

Want to know more?

Website: `http://people.ece.cornell.edu/zivg/`

Email: `goldfeld@cornell.edu`

Office: 322 Rhodes Hall

Spring 2021: **ECE 6970** Statistical Distances for Machine Learning

Thank you!