Information Bottleneck:
Theory, Application, and Controversy

Summarized by Ziyu Ye

Agenda

- Opening the Black-Box of Deep Neural Networks via Information. Tishby and Schwartz-Ziv (2017).
- On the Information Bottleneck Theory of Deep Learning. Saxe et al (2017).
- Deep Variational Information Bottleneck. Alemi et al (2017).

Opening the Black Box of Deep Neural Networks via Information

Agenda

- Overview
- Primer on Information Theory
- The Information Plane
- Optimization v.s. Information Plane
- Rethinking Learning Theory with Information
- Conclusions

Very High-Level Idea of The Paper

1. Information compression can provide a generalization bound in learning theory.

Very High-Level Idea of The Paper

1. Information compression can provide a generalization bound in learning theory.

2. DNN learns in a 2-phase manner:

   Phase 1: Information fitting for target

   Phase 2: Information compression for sample

Very High-Level Idea of The Paper

1. Information compression can provide a generalization bound in learning theory.

2. DNN learns in a 2-phase manner:
   - Phase 1: Information fitting for target
   - Phase 2: Information compression for sample

3. The above two phases matches the 2-stage behavior of SGD optimization:
   - Stage 1: Gradient drift
   - Stage 2: Gradient diffusion

Information Bottleneck: An Illustrative Example

Information Bottleneck: An Illustrative Example

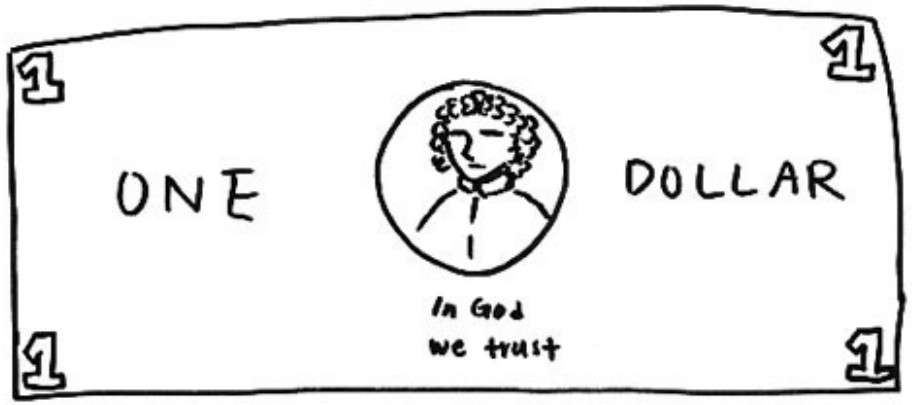


Fig 1a. Drawing of a dollar bill from memory

Fig 1b. How a dollar bill in fact looks like*

* Image source: Epstein, 2016

Information Bottleneck: An Illustrative Example

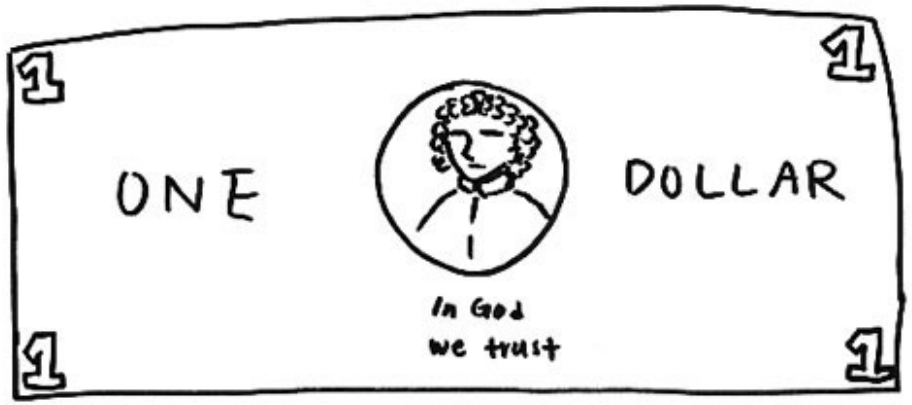


Fig 1a. Drawing of a dollar bill from memory

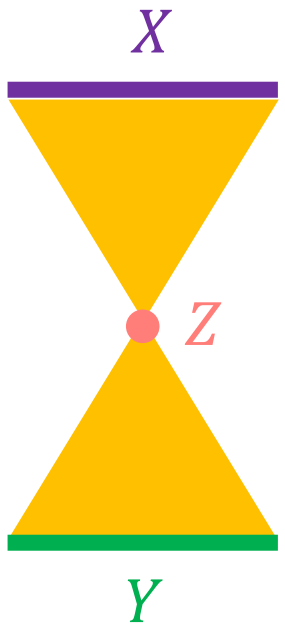
Fig 1b. How a dollar bill in fact looks like*

Goal Find the most relevant features.

Shortcut Forgetting the irrelevant by constraining the information flow.

* Image source: Epstein, 2016

Information Bottleneck: An Illustrative Example



$$\max_{\boldsymbol{\theta}} I(Z, Y; \boldsymbol{\theta})$$

$$\text{s.t. } I(X, Z; \boldsymbol{\theta}) \leq I_c$$

By constraining the information flow, the neural network is forced to learn the most representative features.

Shannon's Information Measure

Entropy

$$H(X) = -\int p(x) \log p(x) dx$$

Joint Entropy

$$H(X,Y) = -\int p(x,y) \log p(x,y) dx dy$$

Conditional Entropy

$$H(X|Y) = -\int p(x,y) \log p(x|y) dx dy$$

KL Divergence

$$KL(X||Y) = \int p(x) \log \frac{p(x)}{p(y)} dx dy$$

Mutual Information

$$I(X;Y) = \int p(x,y) \log \frac{p(x|y)}{p(x,y)} dx dy$$

$$= KL(p(x,y)||p(x)p(y))$$

Properties of Mutual Information

Data Processing Inequality $\forall\, X \to Y \to Z: \;\; I(X;Y) \geq I(X;Z)$

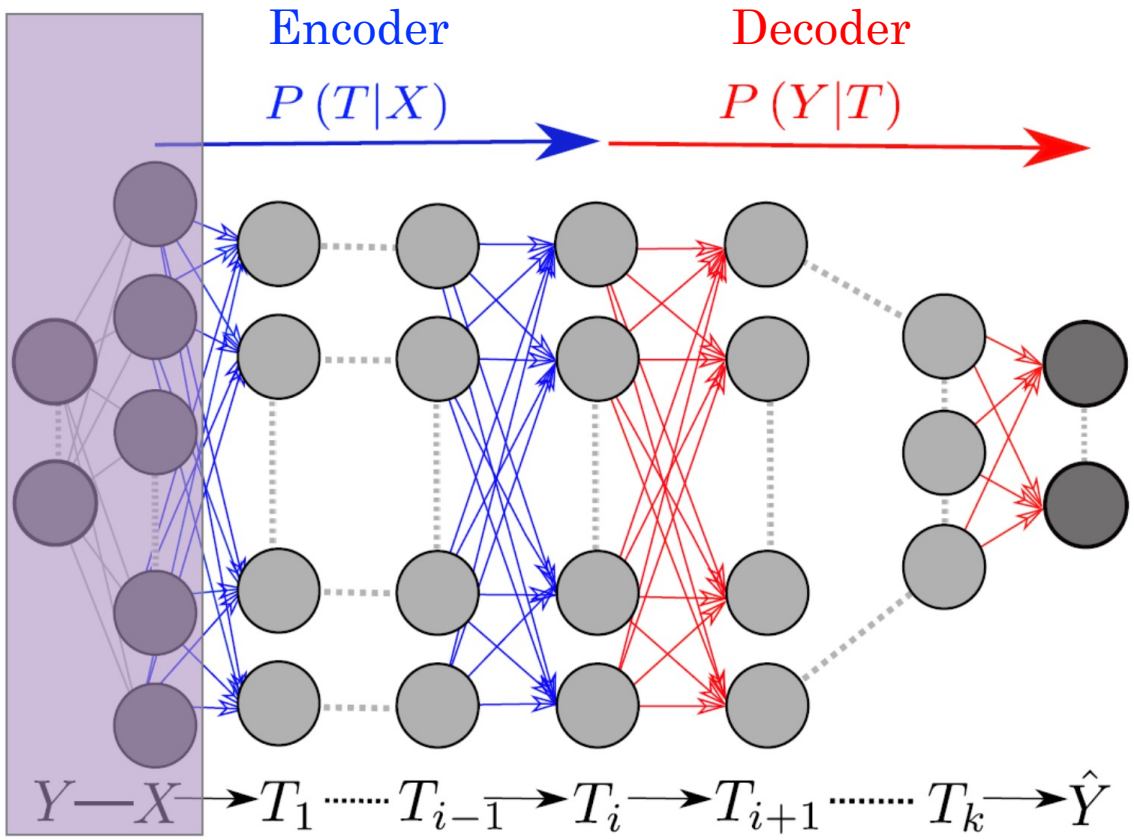
Re-parametrization Invariance $\forall \phi, \psi: \;\; I(X;Y) = I(\phi(X);\psi(Y))$

Properties of Mutual Information

Data Processing Inequality $\forall\, X \to Y \to Z : \;\; I(X;Y) \geq I(X;Z)$

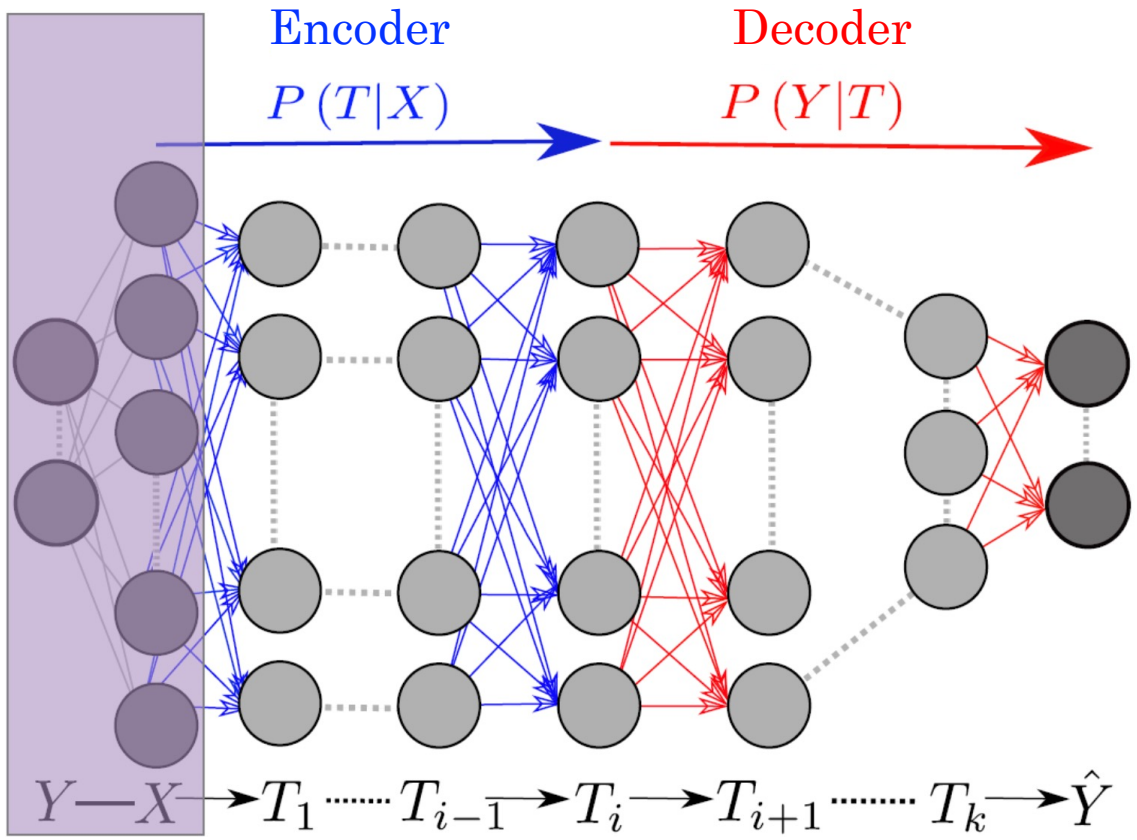
Re-parametrization Invariance $\forall \phi, \psi : \;\; I(X;Y) = I(\phi(X);\psi(Y))$

DNN as Markov Chain of Random Variables



$$I(X;Y) \geq I(T_1;Y) \geq I(T_2;Y) \geq \ldots \geq I(T_k;Y) \geq I(\hat{Y};Y)$$

DNN as Markov Chain of Random Variables



Theorem (Information Plane):

*For large typical **X**, the sample complexity of a DNN is completely determined by the encoder mutual information, **I(X;T)**, of the last hidden layer; the accuracy (generalization error) is determined by the decoder information, **I(T;Y)**, of the last hidden layer.*

$$I(X;Y) \geq I(T_1;Y) \geq I(T_2;Y) \geq \ldots \geq I(T_k;Y) \geq I(\hat{Y};Y)$$

Optimal Representation: Minimal Sufficient Statistics $S(X)$

What is the optimal representation of X w.r.t Y?

Sufficient Statistics $S(X)$

$$I(S(X);Y) = I(X;Y)$$

Optimal Representation: Minimal Sufficient Statistics $S(X)$

What is the optimal representation of X w.r.t Y?

Sufficient Statistics $S(X)$

$$I(S(X);Y) = I(X;Y)$$

Minimal Sufficient Statistics $T(X)$

$$T(X) = \arg \min_{S(X): I(S(X);Y)=I(X;Y)} I(S(X);X)$$

Optimal Representation: Minimal Sufficient Statistics $S(X)$

What is the optimal representation of X w.r.t Y?

Sufficient Statistics $S(X)$

$$I(S(X);Y) = I(X;Y)$$

Minimal Sufficient Statistics $T(X)$

$$T(X) = \arg \min_{S(X): I(S(X);Y) = I(X;Y)} I(S(X);X)$$

Information Bottleneck as an Approximation

$$\min_{p(t|x),p(y|t),p(t)} \{I(X;T) - \beta I(T;Y)\}$$

Optimal Representation: Minimal Sufficient Statistics $S(X)$

What is the optimal representation of X w.r.t Y?

Sufficient Statistics $S(X)$

$$I(S(X);Y) = I(X;Y)$$

Minimal Sufficient Statistics $T(X)$

$$T(X) = \arg \min_{S(X): I(S(X);Y)=I(X;Y)} I(S(X);X)$$

Information Bottleneck as an Approximation

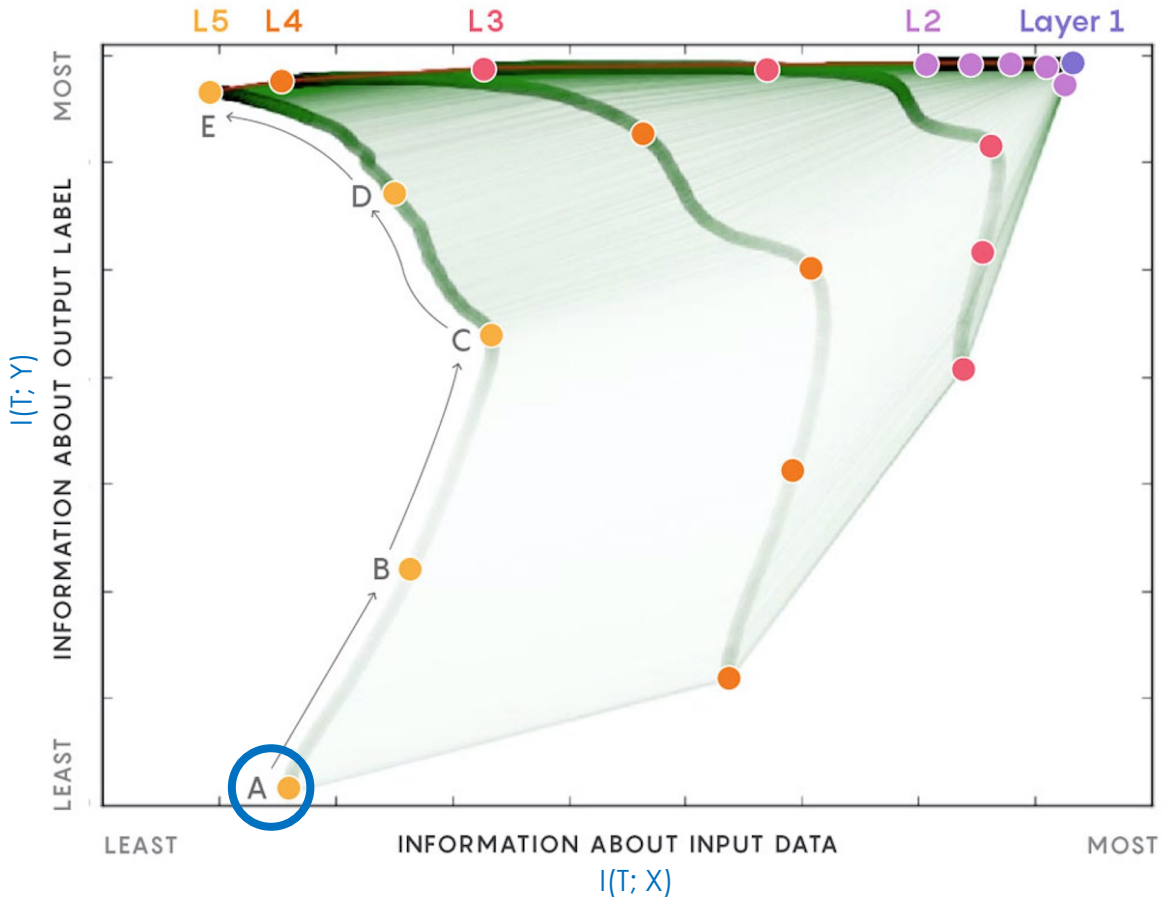
$$\min_{p(t|x),p(y|t),p(t)} \{I(X;T) - \beta I(T;Y)\}$$

$$\begin{cases} p(t|x) = \frac{p(t)}{Z(x;\beta)} \exp\left(-\beta D_{KL}\left[p(y|x) \parallel p(y|t)\right]\right) \\ p(t) = \sum_x p(t|x)\, p(x) \\ p(y|t) = \sum_x p(y|x)\, p(x|t) \ , \end{cases}$$

Information Plane: Evolution of $I(T;X)$ $v.s.$ $I(T;Y)$

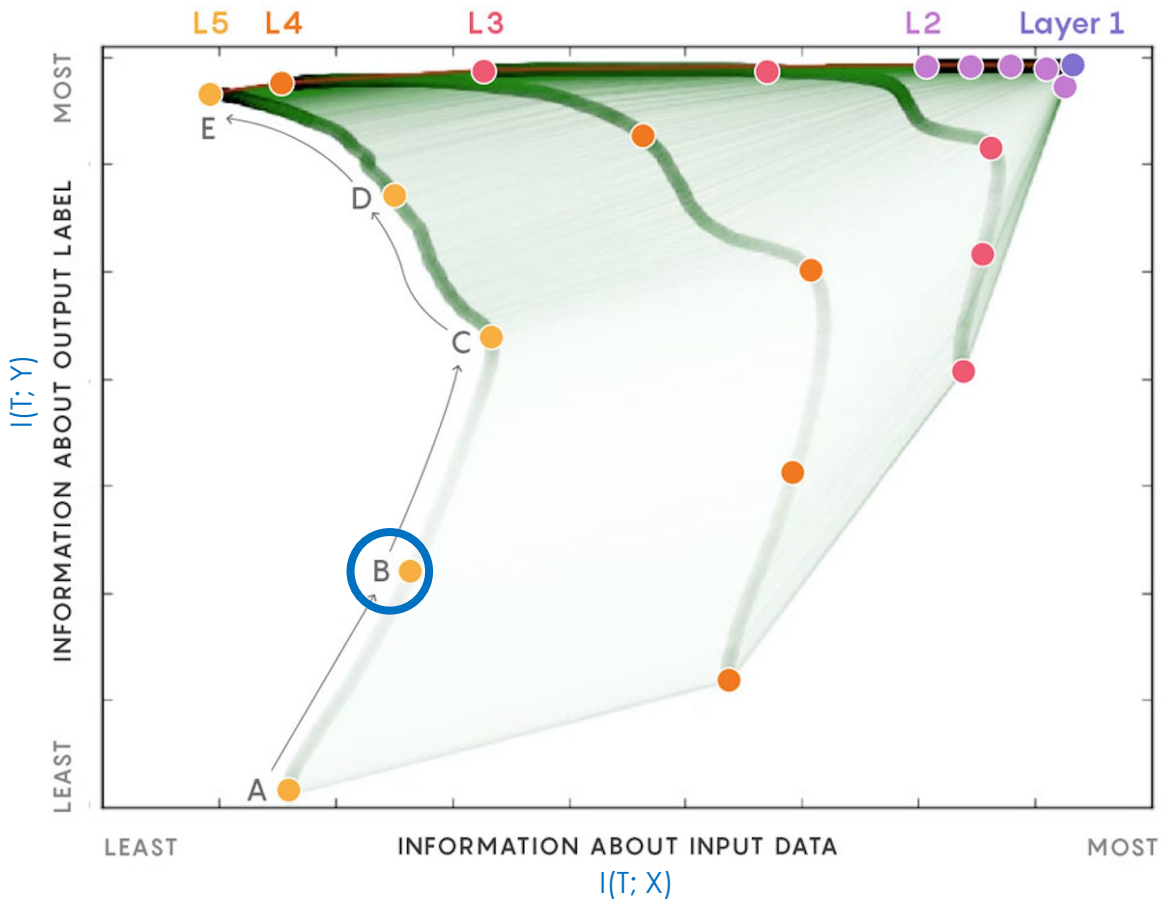
Information Plane: Evolution of $I(T;X)$ $v.s.$ $I(T;Y)$

A Initial State



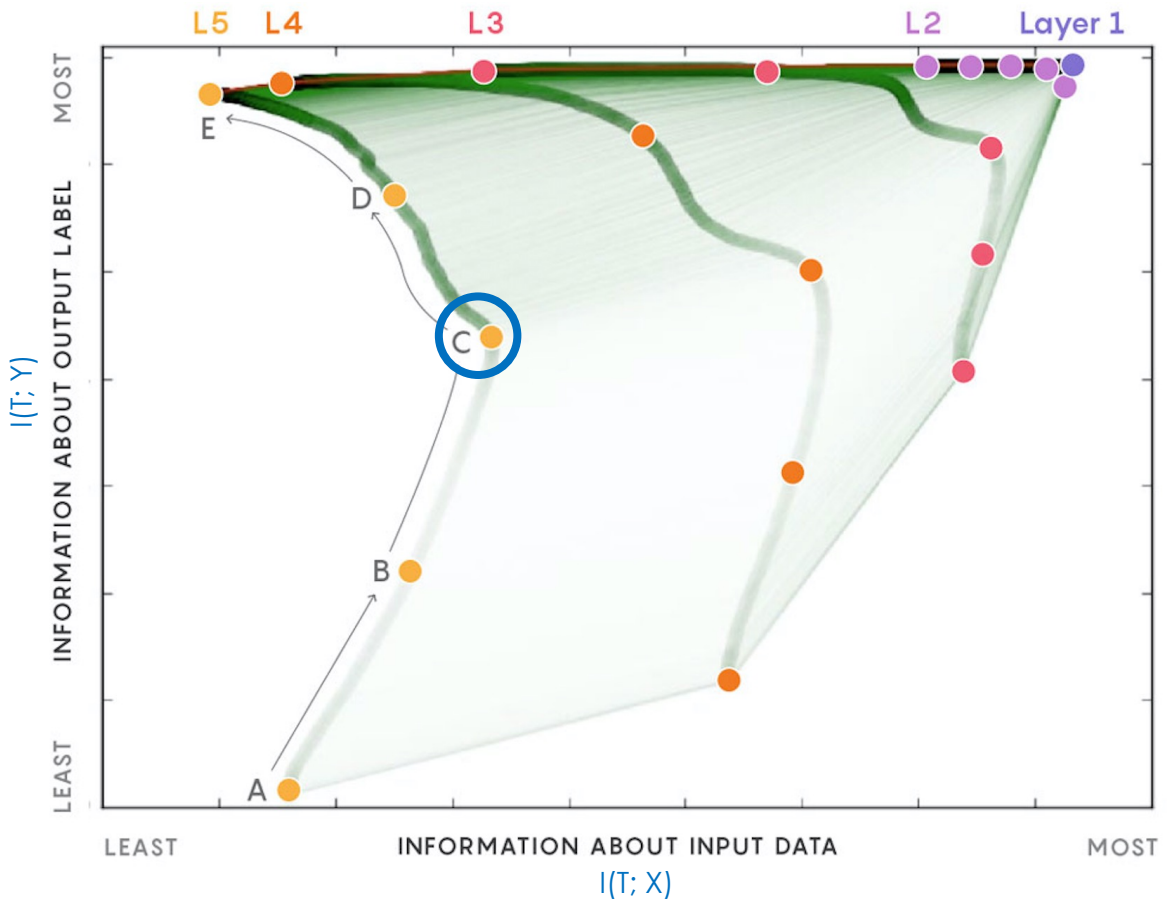
Information Plane: Evolution of $I(T;X)$ $v.s.$ $I(T;Y)$

B Fitting Phase



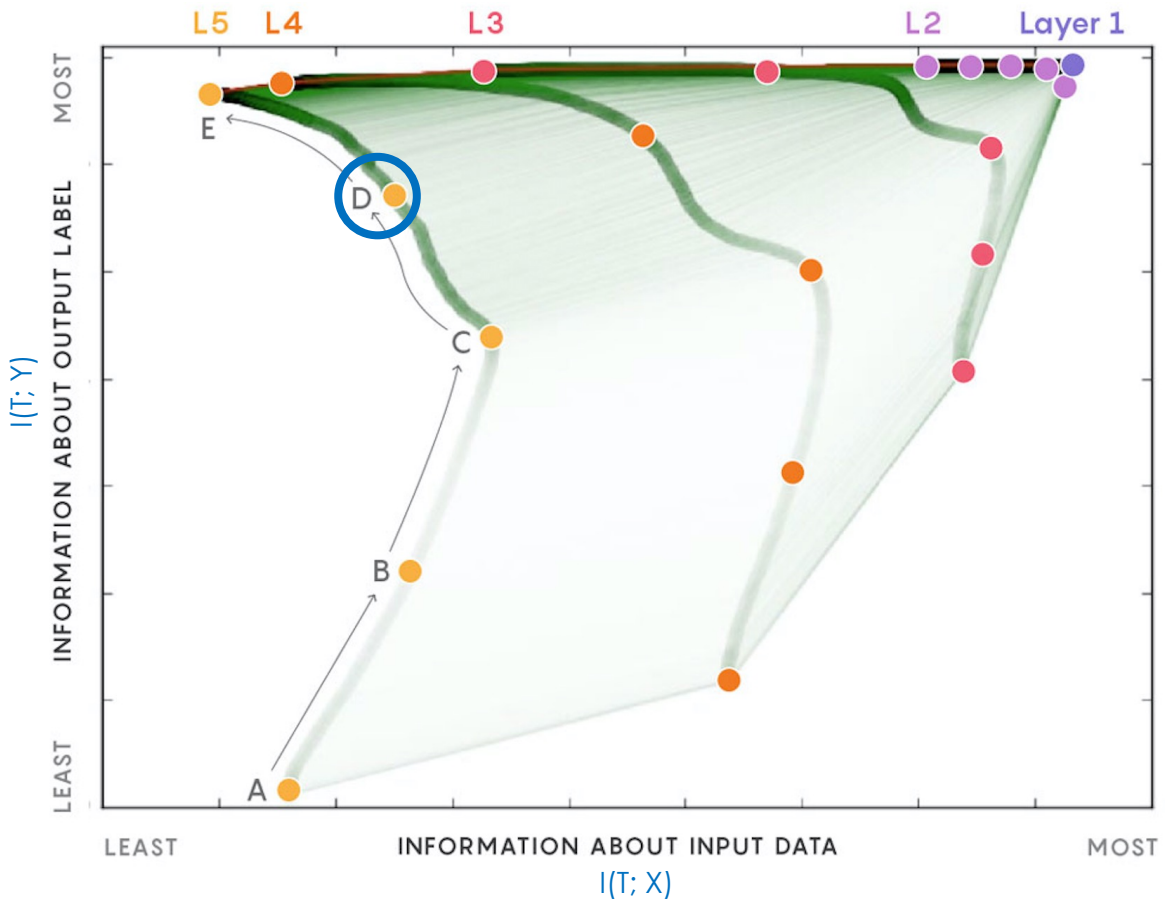
Information Plane: Evolution of $I(T;X)$ $v.s.$ $I(T;Y)$

C Phase Change

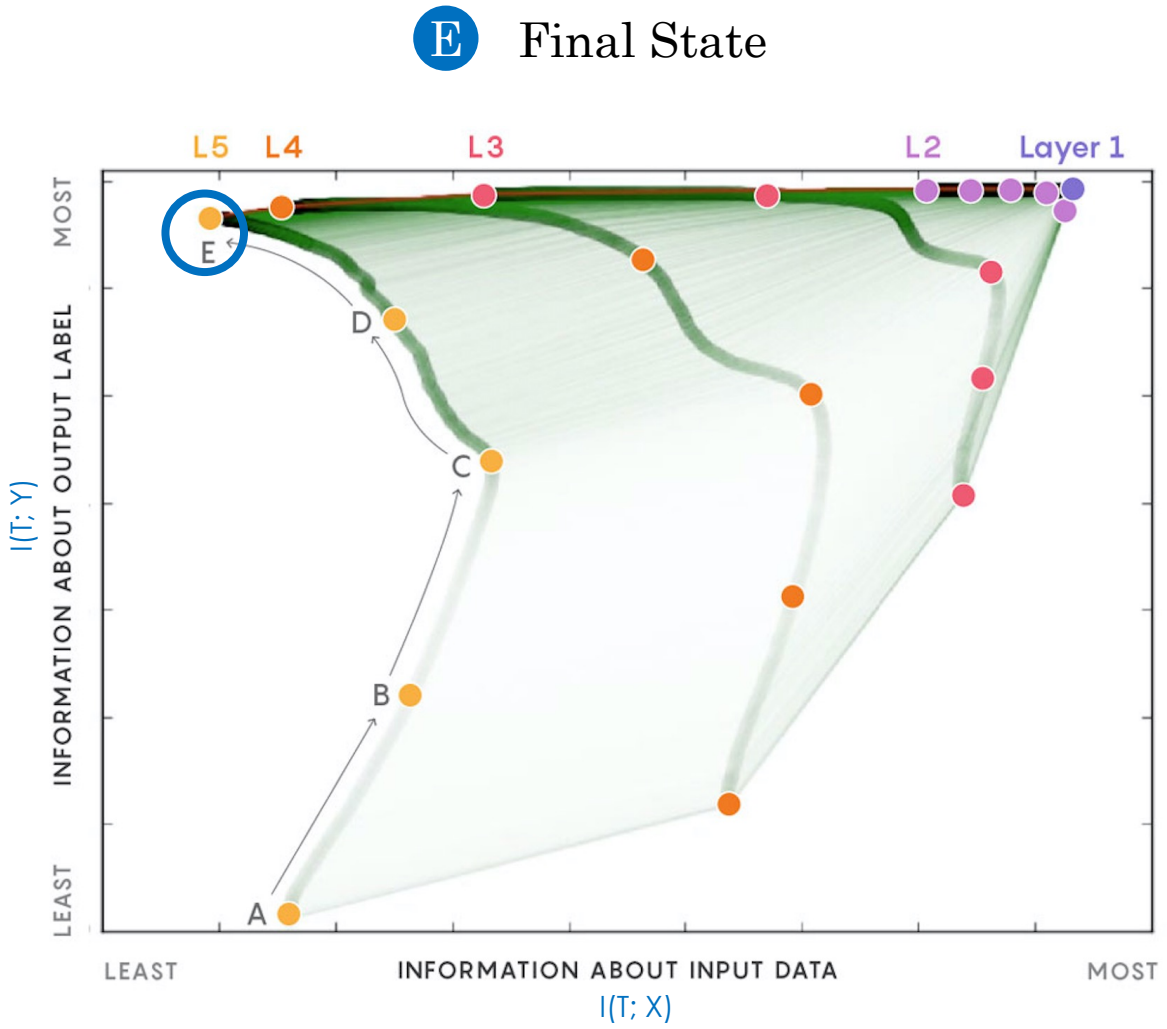


Information Plane: Evolution of $I(T;X)$ $v.s.$ $I(T;Y)$

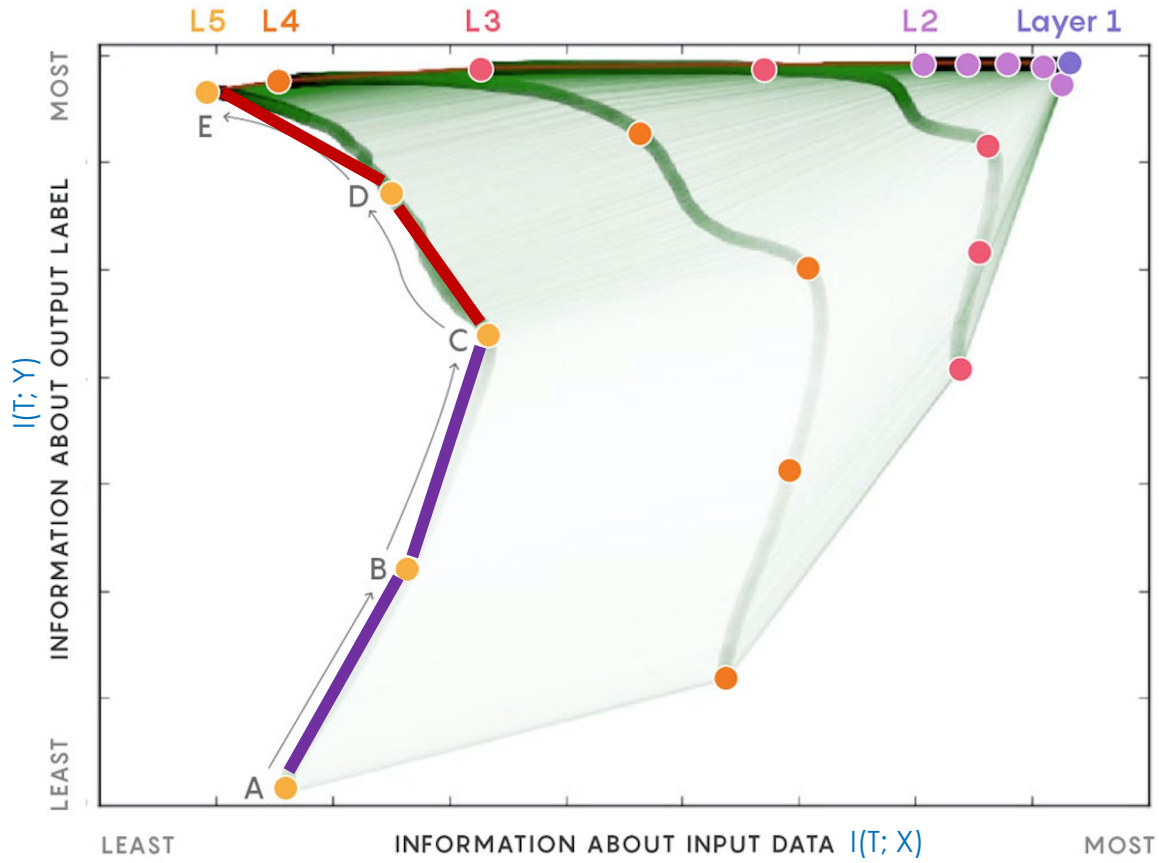
D Compression Phase



Information Plane: Evolution of $I(T;X)$ $v.s.$ $I(T;Y)$



Information Plane: Evolution of $I(T;X)$ $v.s.$ $I(T;Y)$

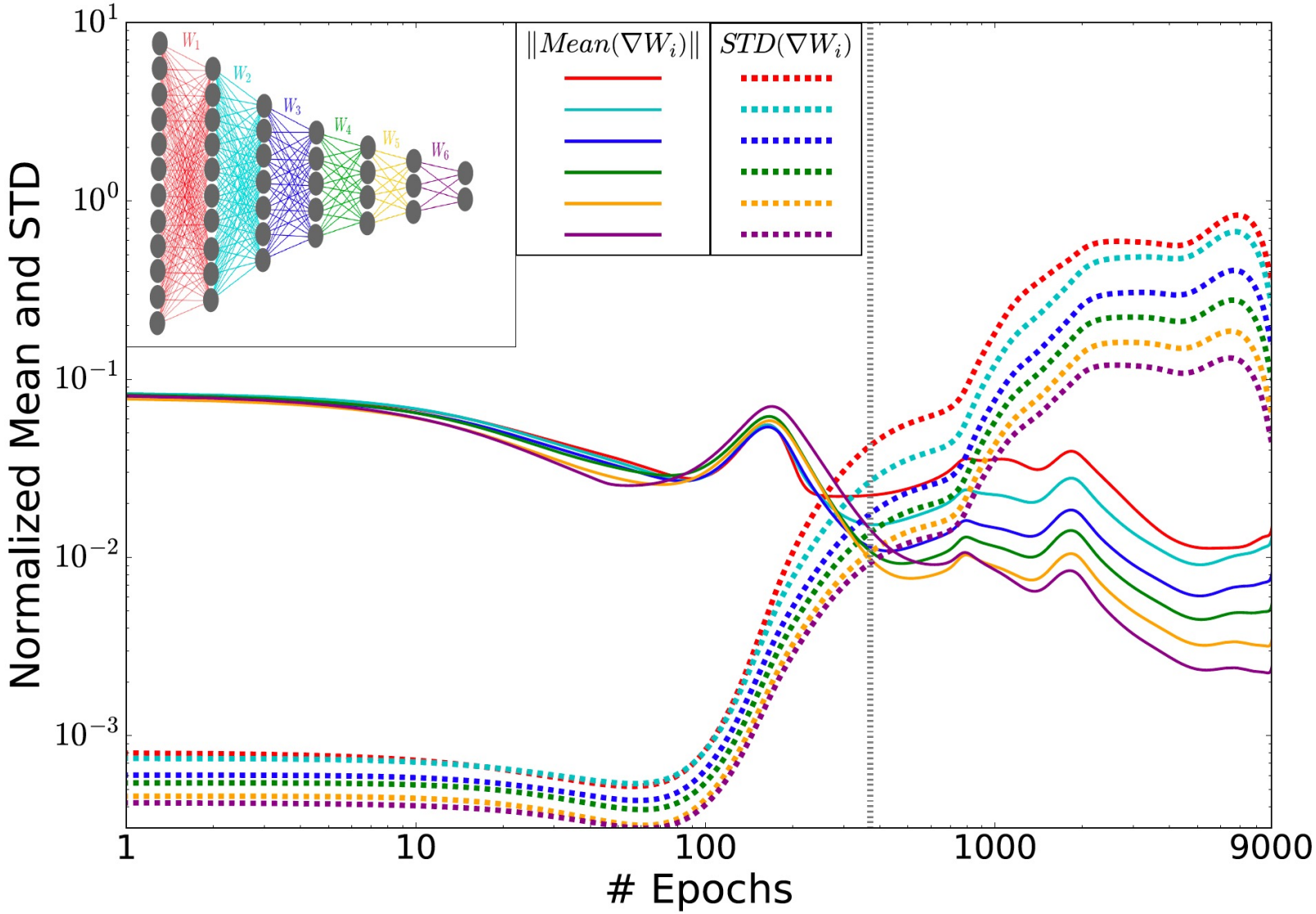


DNN learns in a 2-phase manner:

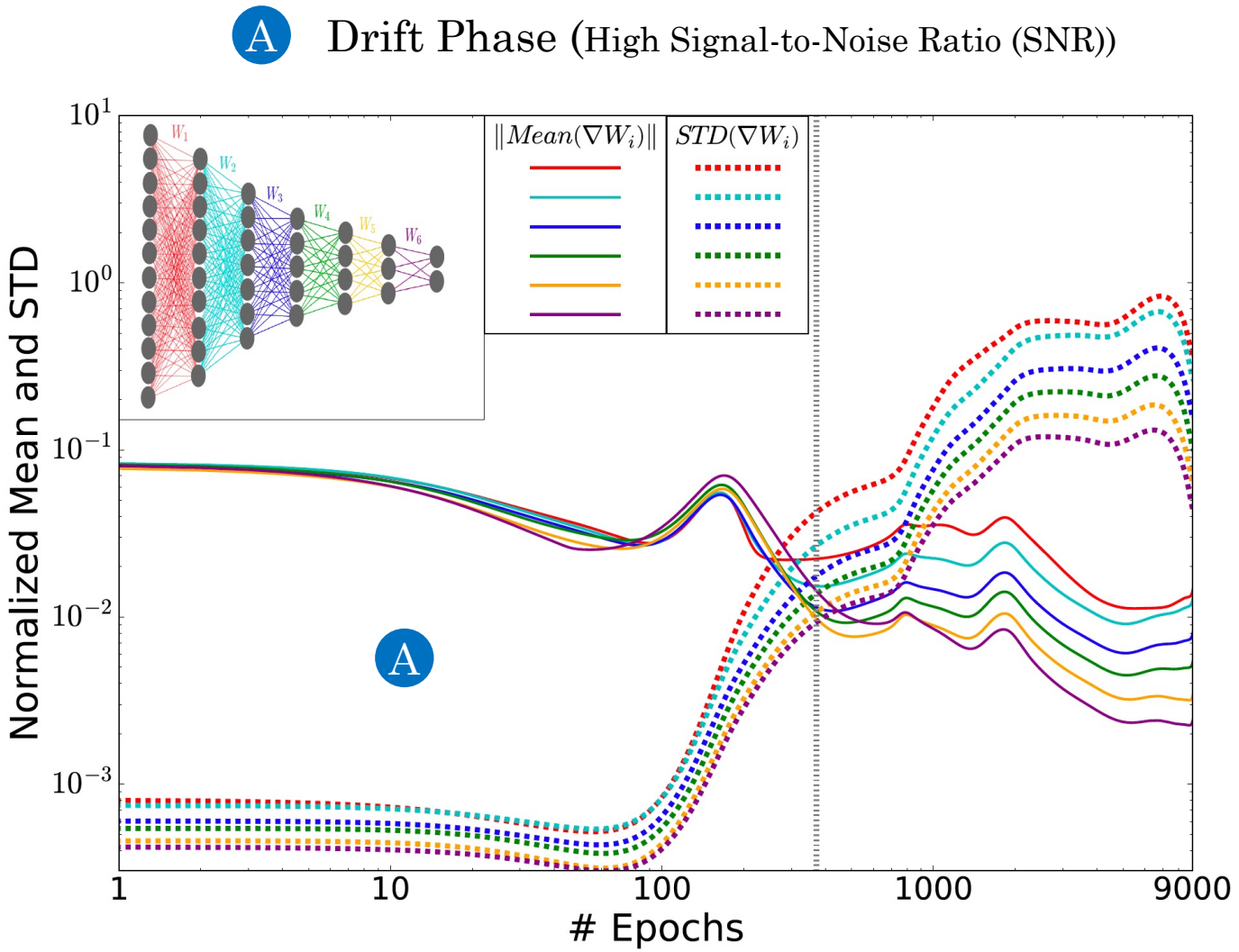
Phase 1: Information fitting for target

Phase 2: Information compression for sample

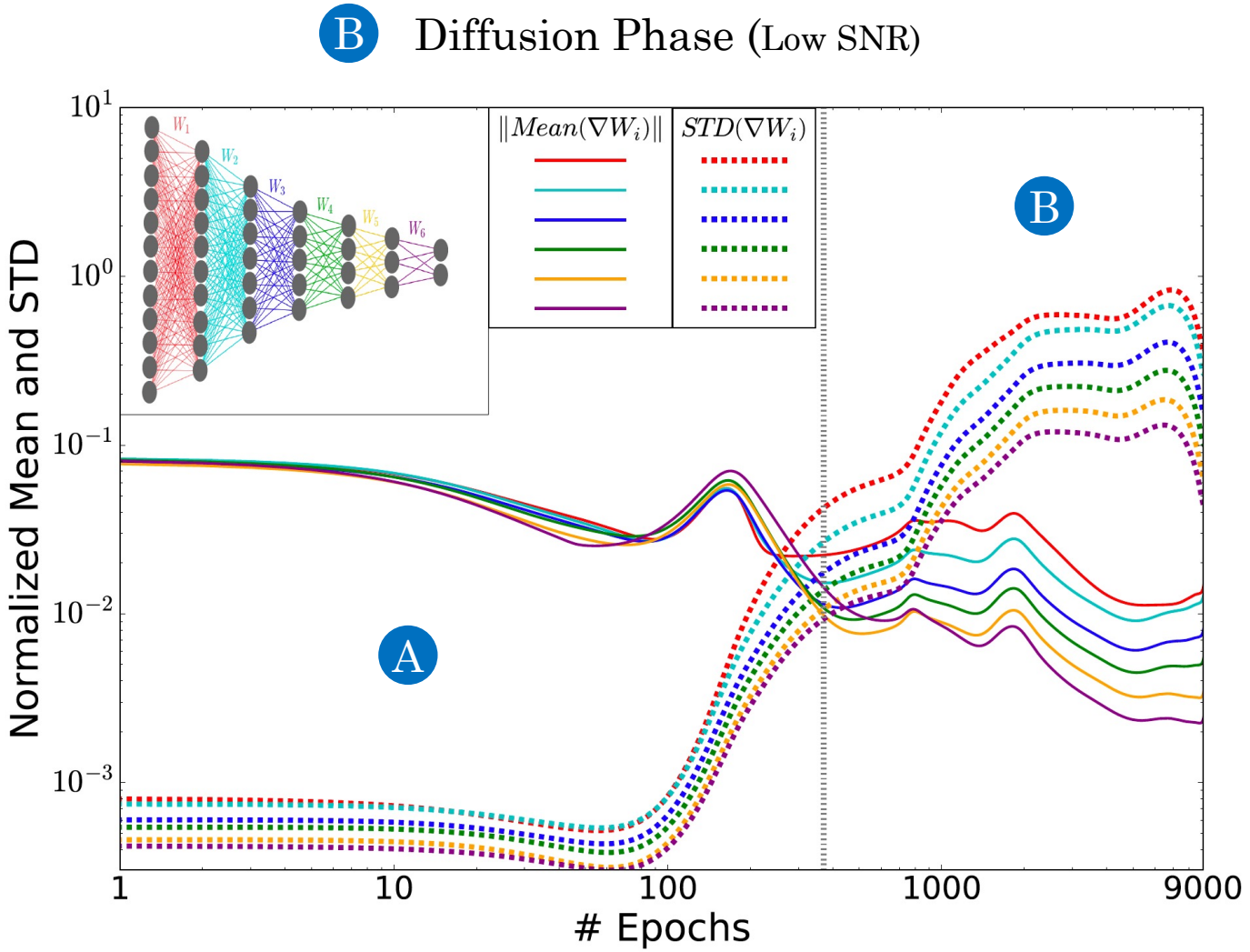
Optimization Process of SGD



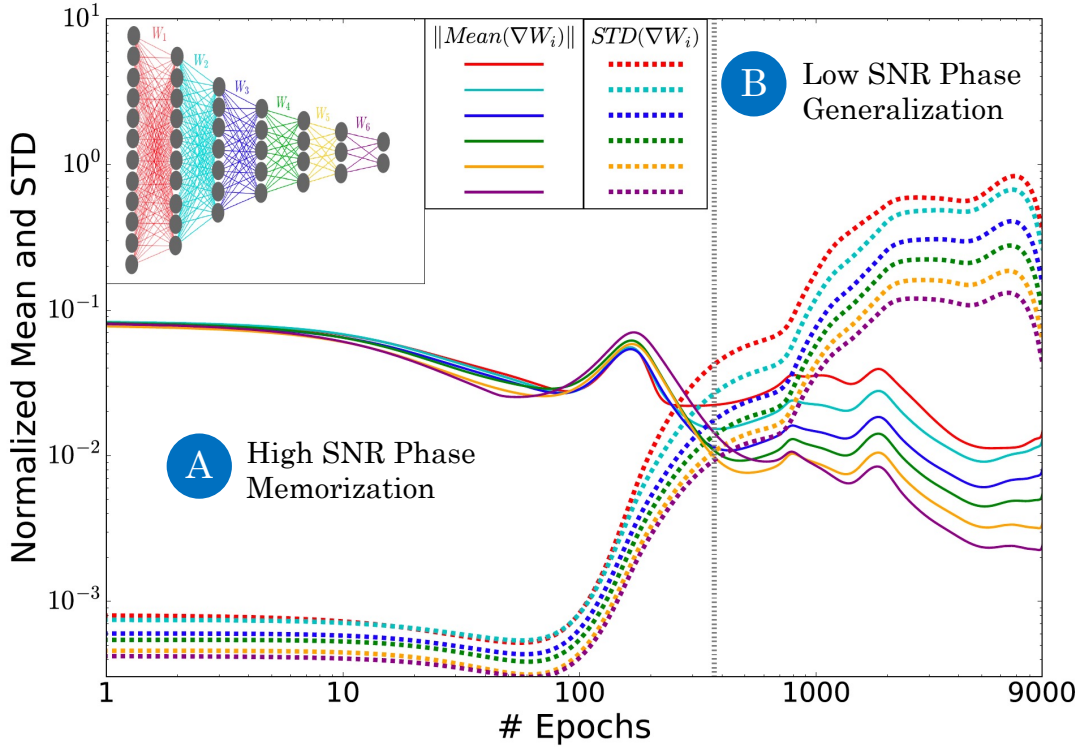
Optimization Process of SGD

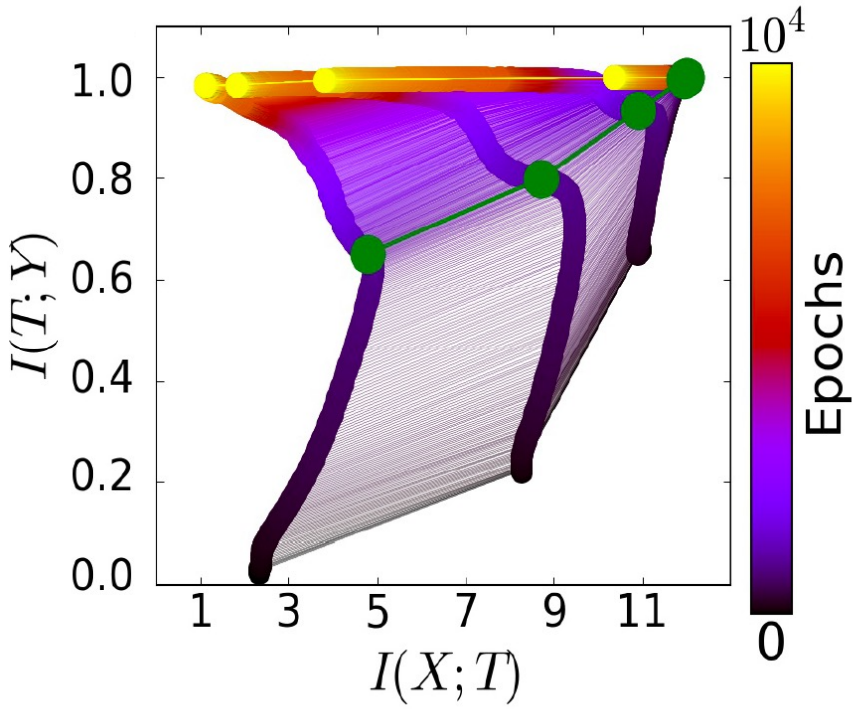


Optimization Process of SGD

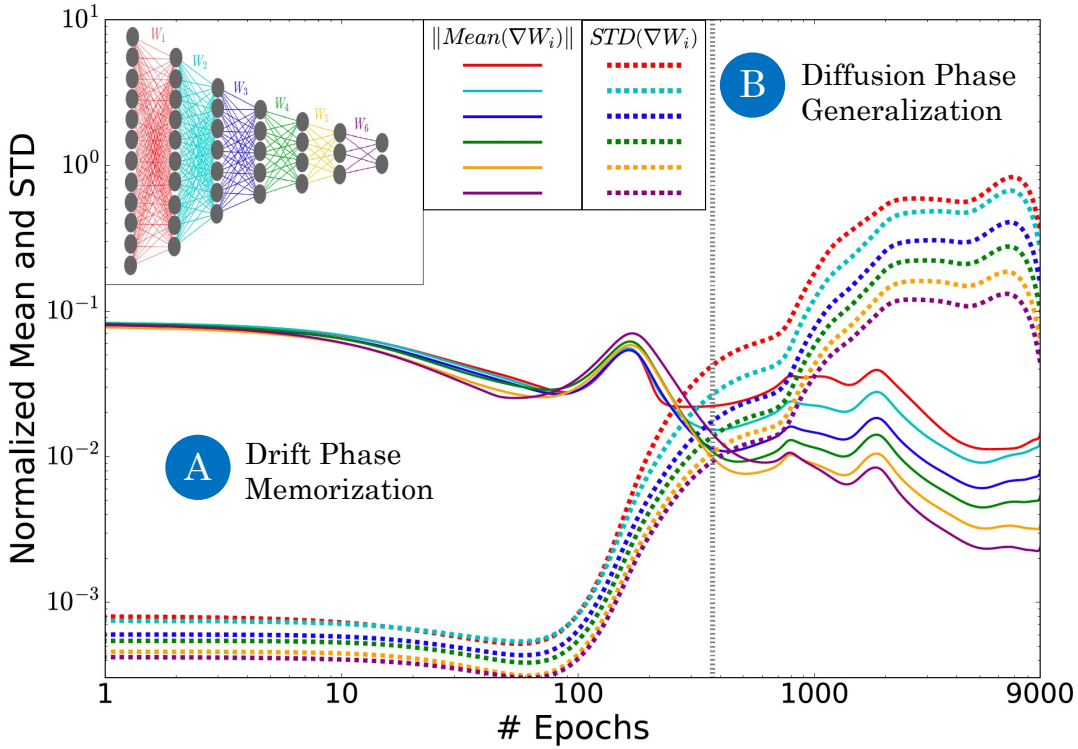


Optimization Process of SGD v.s. Information Plane





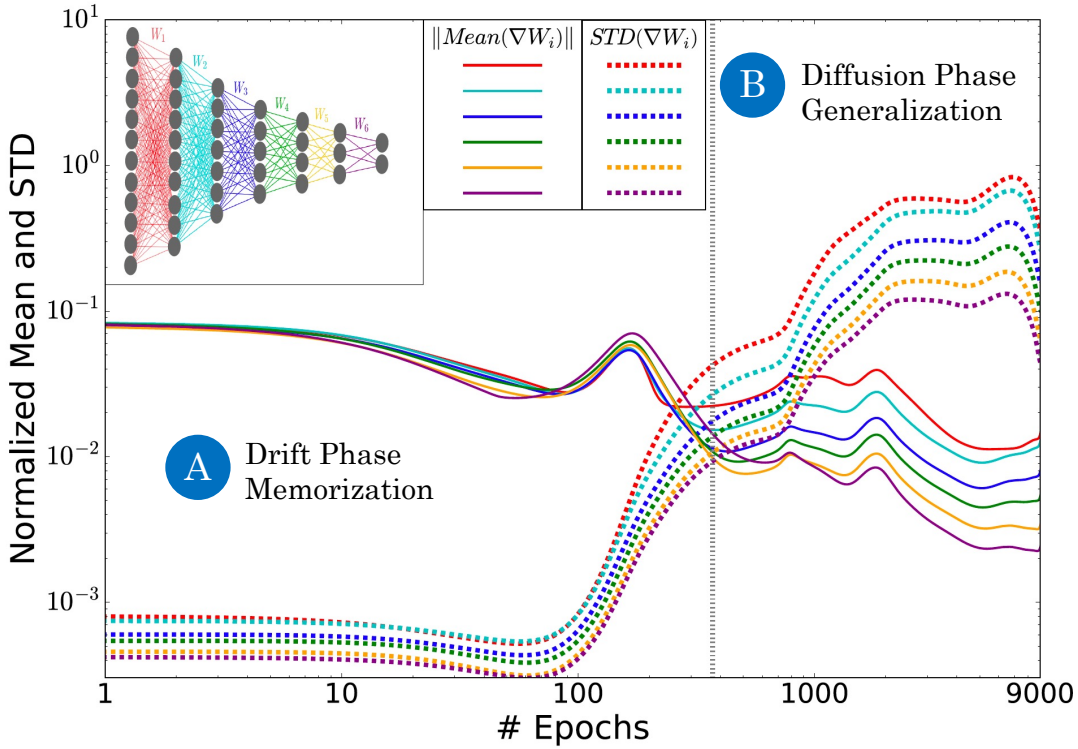
Theoretical Analysis on the Diffusion Phase



1. The diffusion phase works as adding random noise:

$$\frac{\partial W_k}{\partial t} = -\nabla E\left(W_k | X^{(m)}\right) + \beta_k^{-1}\xi(t)$$

Theoretical Analysis on the Diffusion Phase

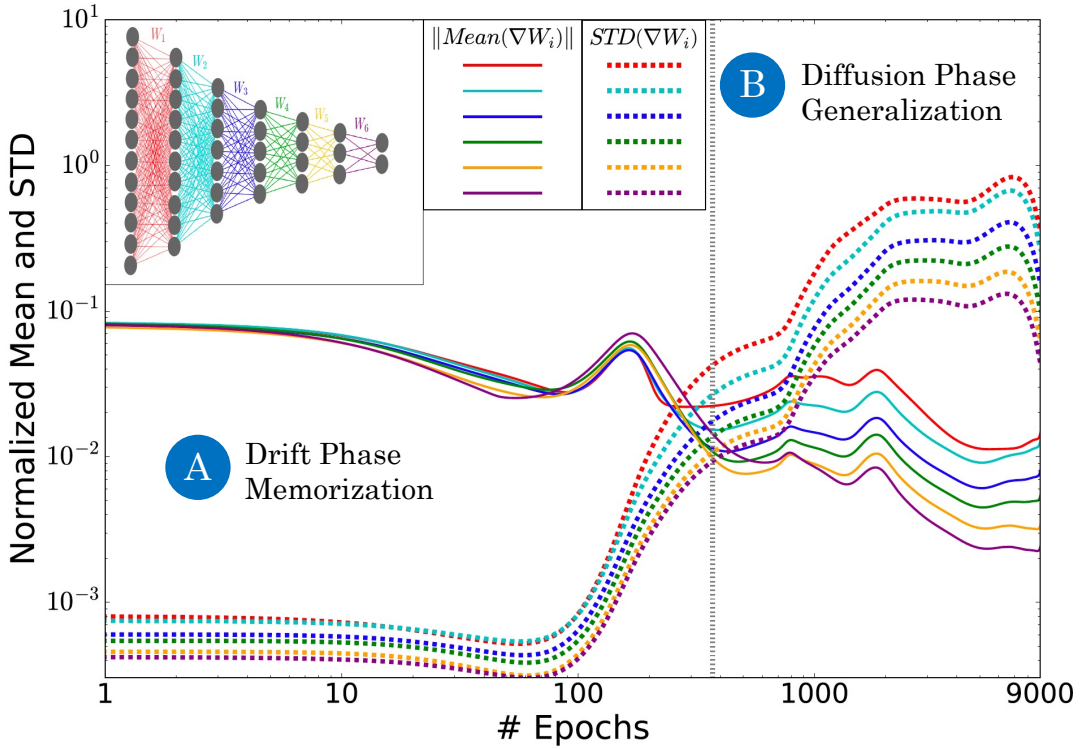


1. The diffusion phase works as adding random noise:

$$\frac{\partial W_k}{\partial t} = -\nabla E\left(W_k | X^{(m)}\right) + \beta_k^{-1}\xi(t)$$

2. The weights evolves like Wienner process.

Theoretical Analysis on the Diffusion Phase



1. The diffusion phase works as adding random noise:

$$\frac{\partial W_k}{\partial t} = -\nabla E\left(W_k|X^{(m)}\right) + \beta_k^{-1}\xi(t)$$

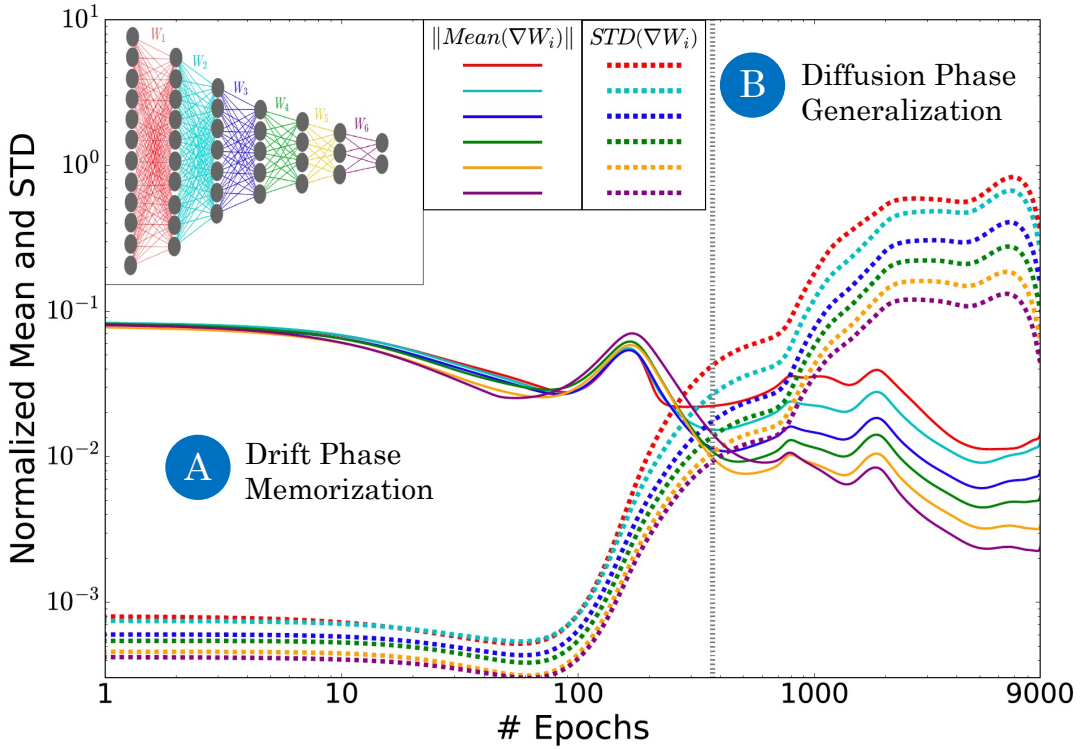
2. The weights evolves like Wienner process.

3. By Focker-Planck equation, the stationary distribution of the weights maximizes its entropy as such.

$$P_{\text{Gibbs}}\left(W_k|X^{(m)}\right) \propto \exp\left(-\beta_k E\left(W_k|X^{(m)}\right)\right)$$

$$P_{Gibbs}\left(X|W_k\right) = P_{Gibbs}\left(X|T_k\right) \propto \exp\left(-\tilde{\beta}_k D_{KL}[p(Y|X)\|p(Y|T)]\right)$$

Theoretical Analysis on the Diffusion Phase



1. The diffusion phase works as adding random noise:

$$\frac{\partial W_k}{\partial t} = -\nabla E\left(W_k|X^{(m)}\right) + \beta_k^{-1}\xi(t)$$

2. The weights evolves like Wienner process.

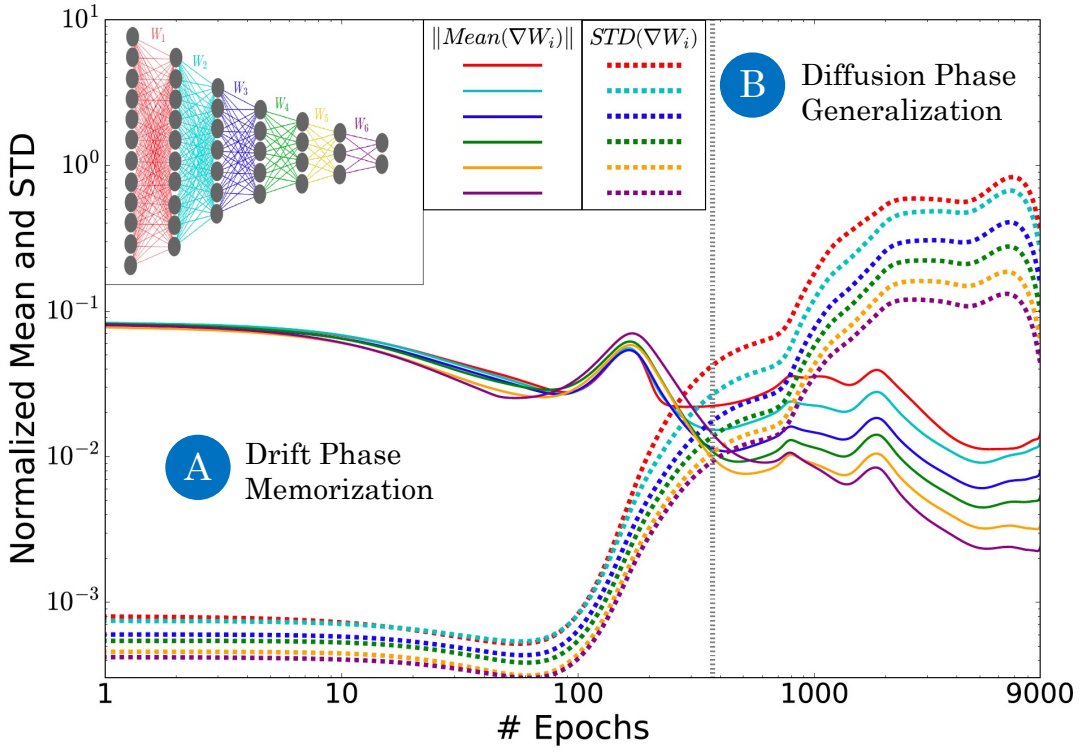
3. By Focker-Planck equation, the stationary distribution of the weights maximizes its entropy as such.

$$P_{\text{Gibbs}}\left(W_k|X^{(m)}\right) \propto \exp\left(-\beta_k E\left(W_k|X^{(m)}\right)\right)$$

$$P_{Gibbs}\left(X|W_k\right) = P_{Gibbs}\left(X|T_k\right) \propto \exp\left(-\tilde{\beta}_k D_{KL}[p(Y|X)\|p(Y|T)]\right)$$

4. That in turn maximize $H\left(X|T_k\right)$.

Theoretical Analysis on the Diffusion Phase



1. The diffusion phase works as adding random noise:

$$\frac{\partial W_k}{\partial t} = -\nabla E\left(W_k|X^{(m)}\right) + \beta_k^{-1}\xi(t)$$

2. The weights evolves like Wienner process.

3. By Focker-Planck equation, the stationary distribution of the weights maximizes its entropy as such.

$$P_{\text{Gibbs}}\left(W_k|X^{(m)}\right) \propto \exp\left(-\beta_k E\left(W_k|X^{(m)}\right)\right)$$

$$P_{Gibbs}\left(X|W_k\right) = P_{Gibbs}\left(X|T_k\right) \propto \exp\left(-\tilde{\beta}_k D_{KL}[p(Y|X)\|p(Y|T)]\right)$$

4. That in turn maximize $H\left(X|T_k\right)$.

5. As $I\left(X;T_k\right) = H(X) - H\left(X|T_k\right)$, that minimizes $I\left(X;T_k\right)$.

Rethinking Generalization Bound

1 "Old" Generalization Bound

$$\varepsilon^2 < \frac{\log|H_\varepsilon| + \log 1/\delta}{2m}$$

ε : generalization error

δ : confidence

m: number of training examples

H_ε: ε-cover of hypothesis class

Rethinking Generalization Bound

1 "Old" Generalization Bound

$$\varepsilon^2 < \frac{\log|H_\varepsilon| + \log 1/\delta}{2m}$$

ε : generalization error

δ : confidence

m: number of training examples

H_ε: ε-cover of hypothesis class

$$|H_\varepsilon| \sim \left(\frac{1}{\varepsilon}\right)^d$$

Rethinking Generalization Bound

1 "Old" Generalization Bound

$$\varepsilon^2 < \frac{\log|H_\varepsilon| + \log 1/\delta}{2m}$$

ε : generalization error
δ : confidence
m: number of training examples
H_ε: ε-cover of hypothesis class

$$|H_\varepsilon| \sim \left(\frac{1}{\varepsilon}\right)^d$$

The "old" bound cannot explain generalizability of deep learning, in which $d \gg m$.

Rethinking Generalization Bound

1 "Old" Generalization Bound

$$\varepsilon^2 < \frac{\log|H_\varepsilon| + \log 1/\delta}{2m}$$

ε : generalization error
δ : confidence
m: number of training examples
H_ε: ε-cover of hypothesis class

$$|H_\varepsilon| \sim \left(\frac{1}{\varepsilon}\right)^d$$

The "old" bound cannot explain generalizability of deep learning, in which $d \gg m$.

2 New Generalization Bound by Input Compression

Rethinking Generalization Bound

1 "Old" Generalization Bound

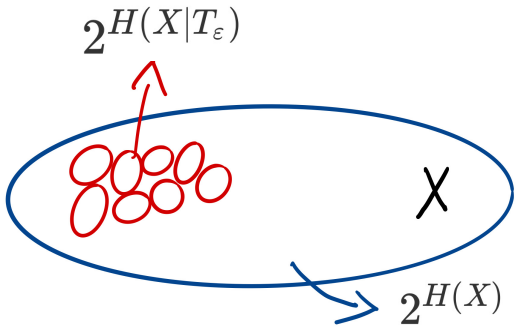
$$\varepsilon^2 < \frac{\log|H_\varepsilon| + \log 1/\delta}{2m}$$

ε : generalization error
δ : confidence
m: number of training examples
H_ε: ε-cover of hypothesis class

$$|H_\varepsilon| \sim \left(\frac{1}{\varepsilon}\right)^d$$

The "old" bound cannot explain generalizability of deep learning, in which $d \gg m$.

2 New Generalization Bound by Input Compression



$$|H_\varepsilon| \sim 2^{|X|} \rightarrow 2^{|T_\varepsilon|}$$

T_ε: ε-partition of input variable X

Rethinking Generalization Bound

1 "Old" Generalization Bound

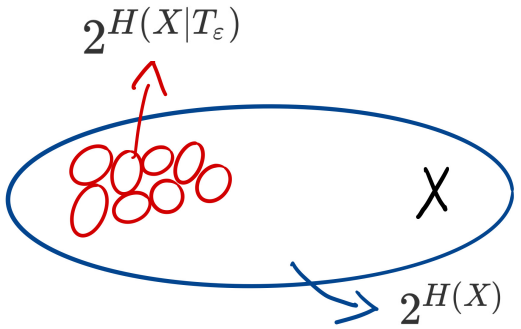
$$\varepsilon^2 < \frac{\log|H_\varepsilon| + \log 1/\delta}{2m}$$

ε : generalization error
δ : confidence
m: number of training examples
H_ε: ε-cover of hypothesis class

$$|H_\varepsilon| \sim \left(\frac{1}{\varepsilon}\right)^d$$

The "old" bound cannot explain generalizability of deep learning, in which $d \gg m$.

2 New Generalization Bound by Input Compression



$$|H_\varepsilon| \sim 2^{|X|} \rightarrow 2^{|T_\varepsilon|}$$

T_ε: ε-partition of input variable X

$$|T_\varepsilon| \sim 2^{I(T_\varepsilon;X)}$$

Rethinking Generalization Bound

1 "Old" Generalization Bound

$$\varepsilon^2 < \frac{\log|H_\varepsilon| + \log 1/\delta}{2m}$$

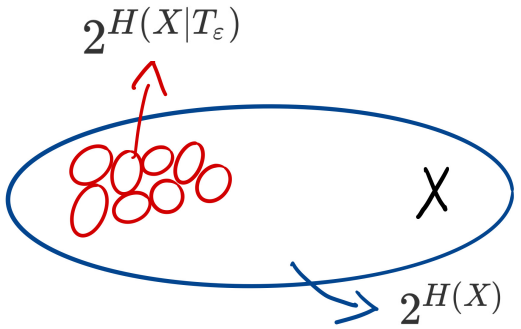
ε : generalization error
δ : confidence
m: number of training examples
H_ε: ε-cover of hypothesis class

$$|H_\varepsilon| \sim \left(\frac{1}{\varepsilon}\right)^d$$

The "old" bound cannot explain generalizability of deep learning, in which $d \gg m$.

2 New Generalization Bound by Input Compression

$$\varepsilon^2 < \frac{2^{I(T_\varepsilon;X)} + \log 1/\delta}{2m}$$



$$|H_\varepsilon| \sim 2^{|X|} \rightarrow 2^{|T_\varepsilon|}$$

T_ε: ε-partition of input variable X

$$|T_\varepsilon| \sim 2^{I(T_\varepsilon;X)}$$

Conclusion

1. Information compression can provide a generalization bound in learning theory.

2. DNN learns in a 2-phase manner:
   - Phase 1: Information fitting for target
   - Phase 2: Information compression for sample

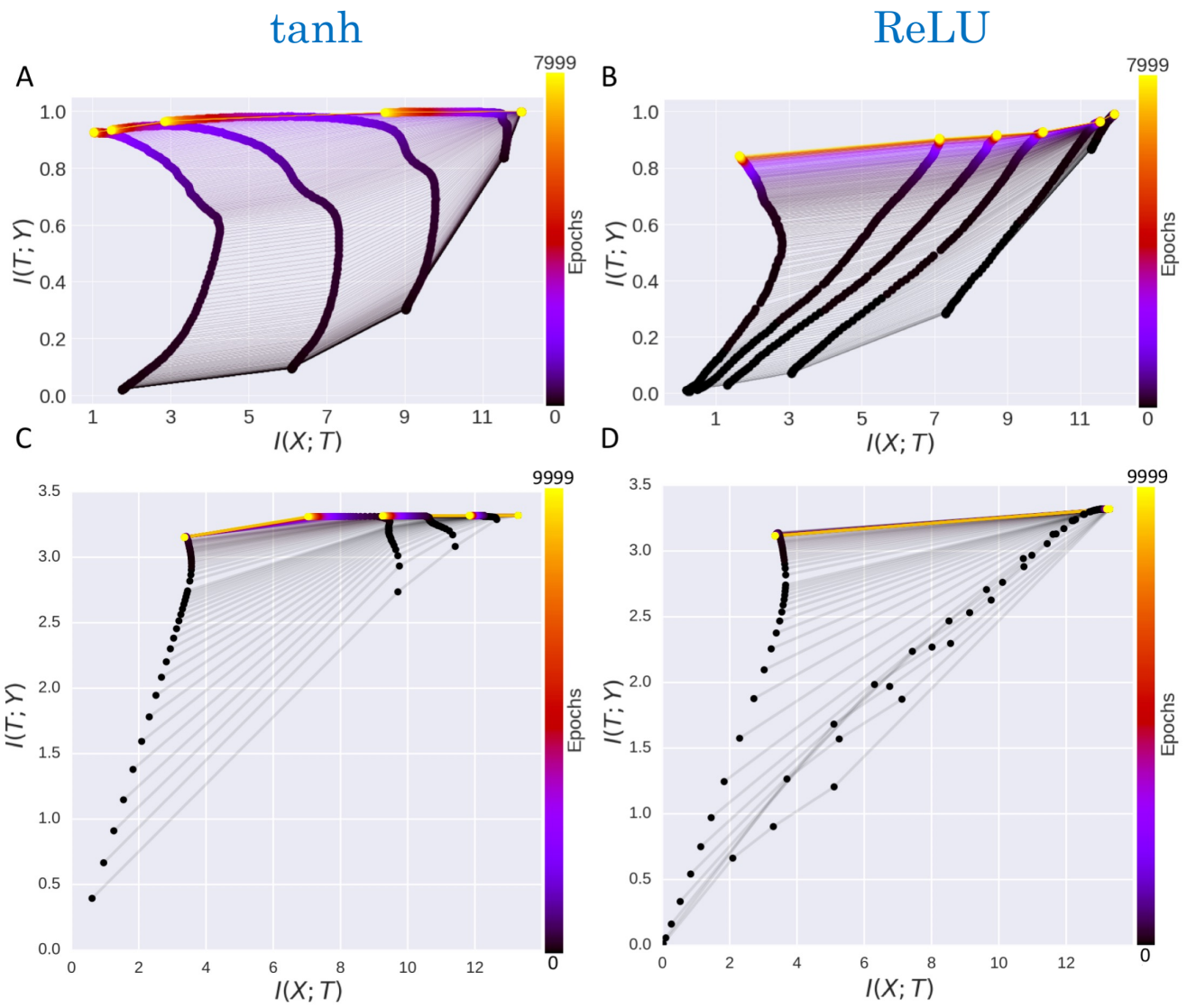
3. The above two phases matches the 2-stage behavior of SGD optimization:
   - Stage 1: Gradient drift
   - Stage 2: Gradient diffusion

On the Information Bottleneck Theory of Deep Learning

Agenda

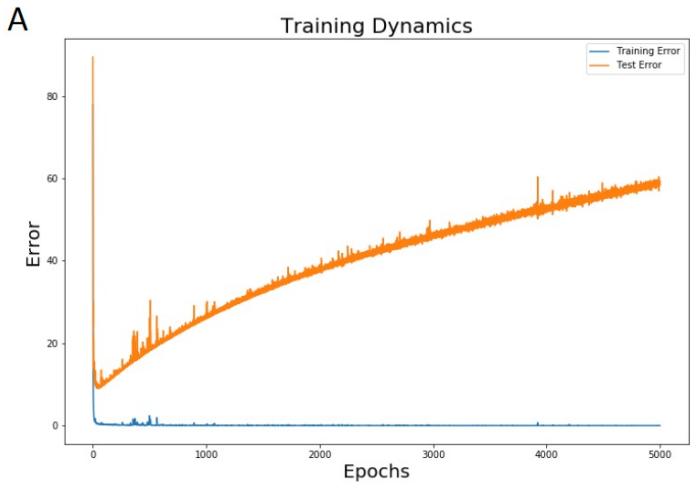
- Overview
- Compression Depends on Activation
- Generalizability v.s. Information Plane
- Optimization v.s. Information Plane
- Conclusions

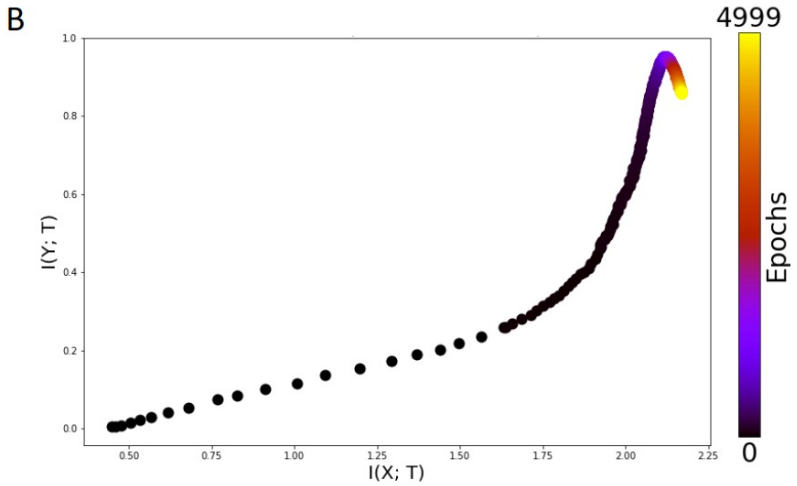
Tanh v.s. ReLU


tanh
ReLU
Synthetic Dataset
MNIST
A
B
C
D
7999
9999
Epochs
I(T; Y)
I(X; T)

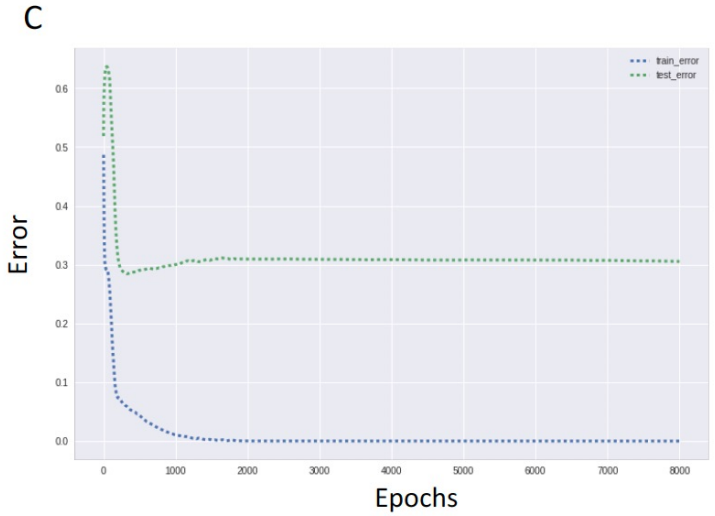
Overfitting: Linear v.s. Nonlinear

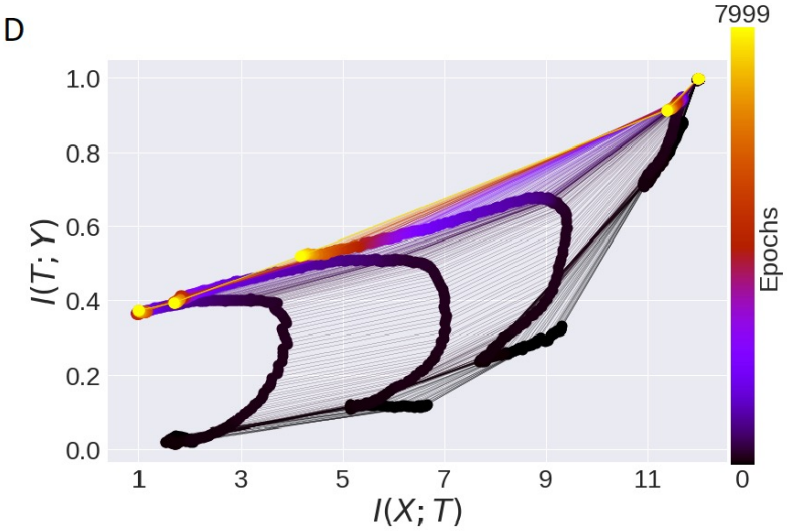
Linear



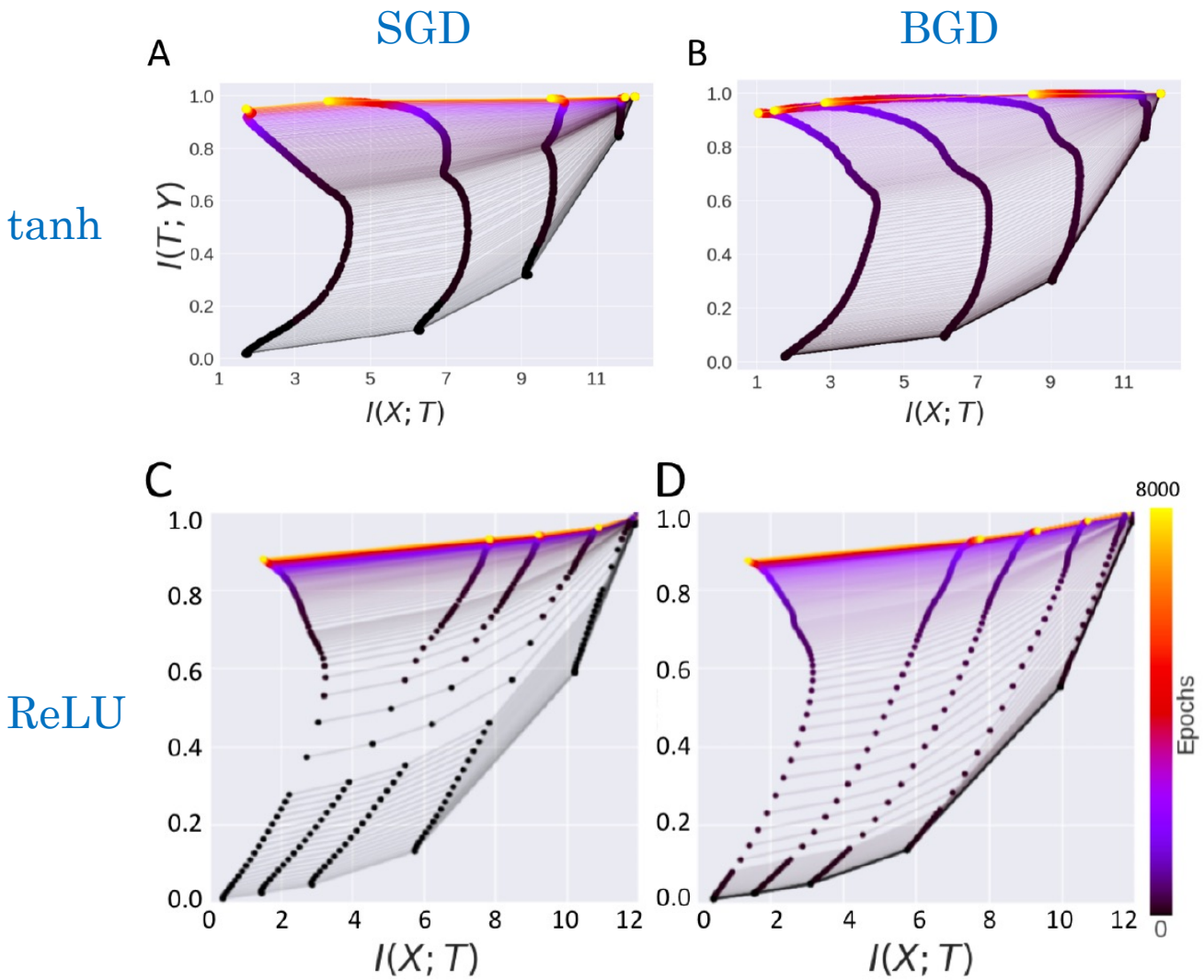


Non-Linear

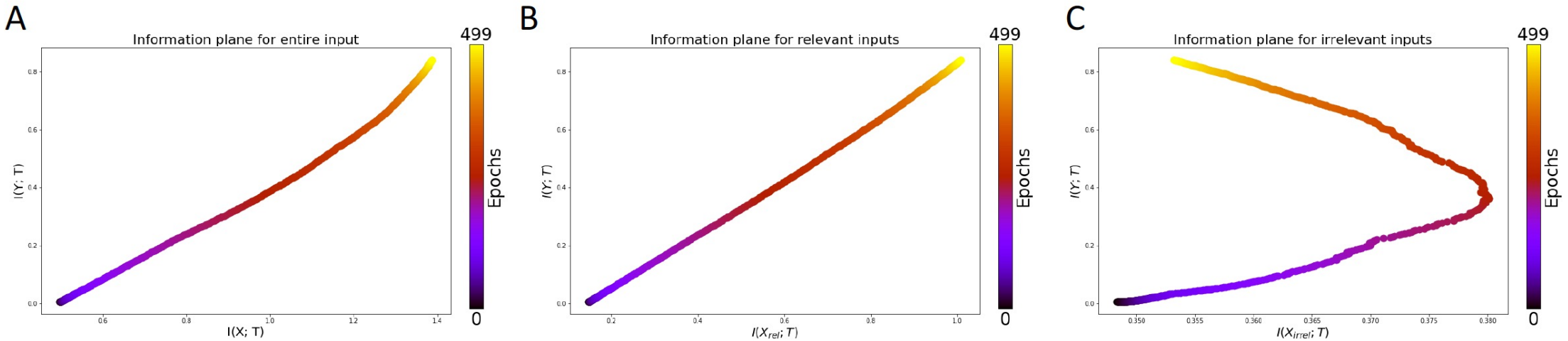


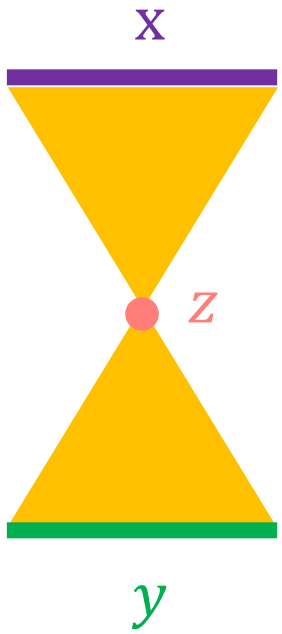


Stochastic v.s. Non-Stochastic



Compression: Relevance v.s. Irrelevant



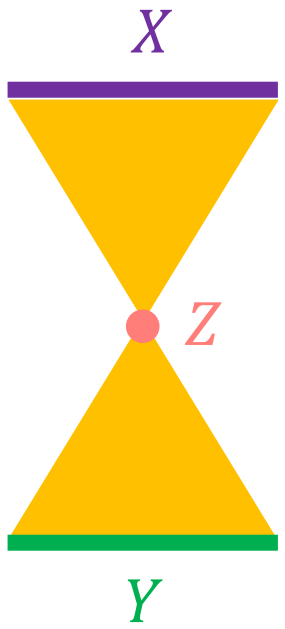


Deep Variational Information Bottleneck

Agenda

- Overview
- Primer on Variational Inference
- Variational Information Bottleneck (VIB)
- Experiment Results for VIB
- Conclusions

Information Bottleneck as a Regularizer for DNN



$$\max_{\boldsymbol{\theta}} I(Z, Y; \boldsymbol{\theta})$$

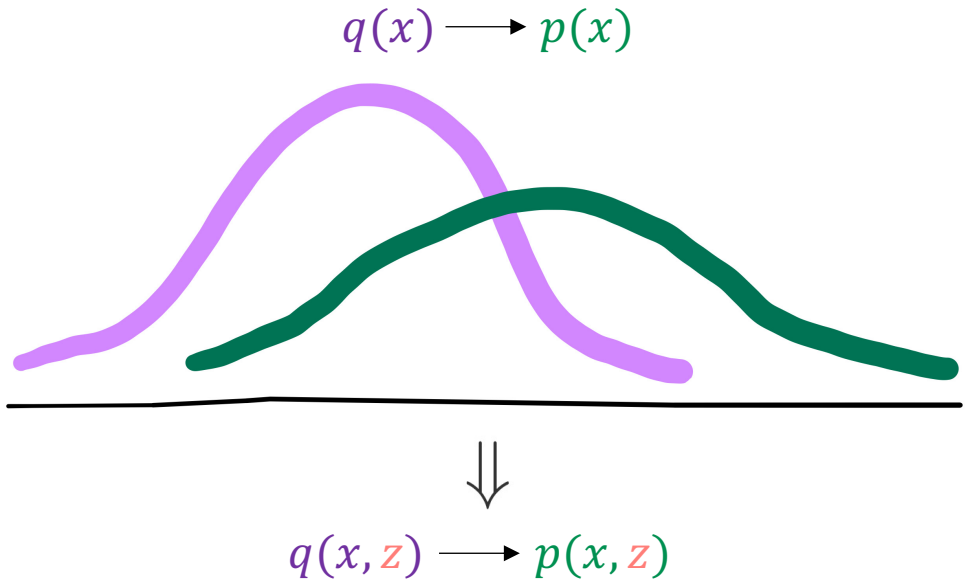
$$\text{s.t. } I(X, Z; \boldsymbol{\theta}) \leq I_c$$

By constraining the information flow, the neural network is forced to learn the most representative features.

- Better generalizability
- Robustness to attacks

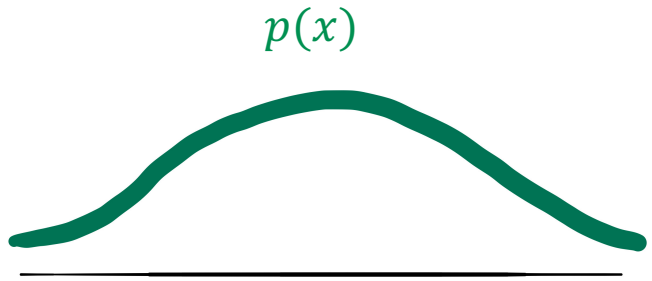
Primer on Variational Inference

Variational inference is to approximate intractable distribution with latent variables.



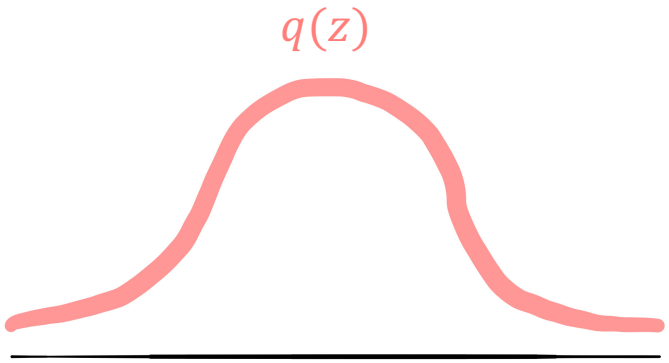
Variational Inference for Generative Models

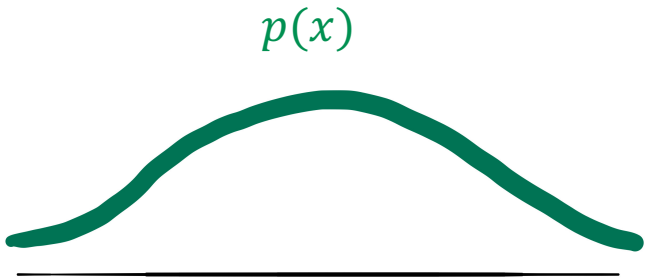
Variational inference is to approximate intractable distribution with latent variables.



Variational Inference for Generative Models

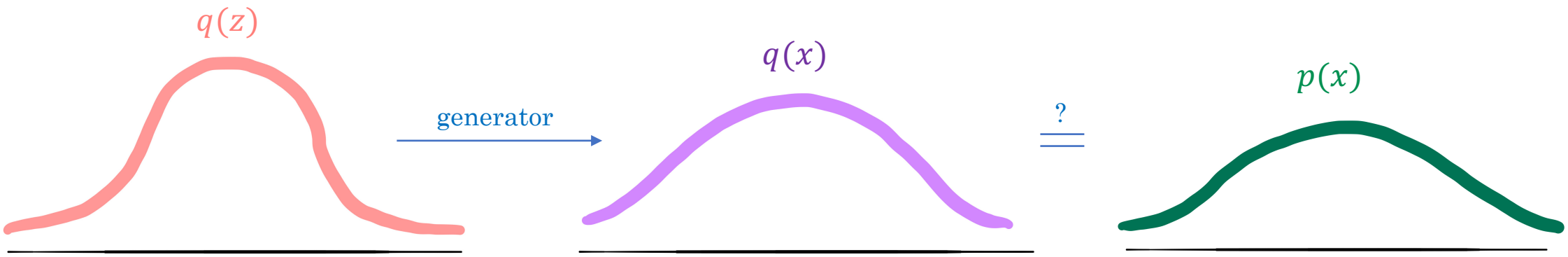
Variational inference is to approximate intractable distribution with latent variables.





Variational Inference for Generative Models

Variational inference is to approximate intractable distribution with latent variables.



Derivation for Variational Autoencoder (VAE)

$p(x)$: Evidence probability for the sample $\{x_i\}_1^N$

$q(x)$: Proposed probability to approximate $p(x)$

Step 1: Approximating by minimizing KL divergence: $\min KL(p(x) \,||\, q(x))$

Derivation for VAE

$p(x)$: Evidence probability for the sample $\{x_i\}_1^N$

$q(x)$: Proposed probability to approximate $p(x)$

Step 1: Approximating by minimizing KL divergence: $\min KL(p(x) \,||\, q(x))$

Step 2: Introducing latent variable z, by defining:

$$p(x,z) = p(x)p(z|x)$$

$$q(x,z) = q(z)q(x|z)$$

Derivation for VAE

$p(x)$: Evidence probability for the sample $\{x_i\}_1^N$

$q(x)$: Proposed probability to approximate $p(x)$

$q(z)$: Pre-defined prior for the latent vector z

$p(z|x)$: Posterior of z given x.

$q(x|z)$: Posterior of x given z.

Step 1: Approximating by minimizing KL divergence: $\min KL(p(x) \,||\, q(x))$

Step 2: Introducing latent variable z, by defining:

$$p(x,z) = p(x)p(z|x)$$

$$q(x,z) = q(z)q(x|z)$$

Derivation for VAE

$p(x)$: Evidence probability for the sample $\{x_i\}_1^N$

$q(x)$: Proposed probability to approximate $p(x)$

$q(z)$: Pre-defined prior for the latent vector z

$p(z|x)$: Posterior of z given x.

$q(x|z)$: Posterior of x given z.

Step 1: Approximating by minimizing KL divergence: $\min KL(p(x) \,||\, q(x))$

Step 2: Introducing latent variable z, such that $p(x,z) = p(x)p(z|x)$ and $q(x,z) = q(z)q(x|z)$.

Step 3: Show that $KL(p(x) \,||\, q(x))$ is upper bounded by $KL(p(x,z)||\, q(x,z))$:

$$KL(p(x,z)||\, q(x,z)) = KL\big(p(x) \,||\, q(x)\big) + \int p(x) KL\big(p(z|x) \,||\, q(z|x)\big) dx$$
$$\geq KL\big(p(x) \,||\, q(x)\big)$$

Derivation for VAE

$p(x)$: Evidence probability for the sample $\{x_i\}_1^N$

$q(x)$: Proposed probability to approximate $p(x)$

$q(z)$: Pre-defined prior for the latent vector z

$p(z|x)$: Posterior of z given x.

$q(x|z)$: Posterior of x given z.

Step 4: Reformulate the goal:

$$\min KL(p(x,z)||\, q(x,z))$$

$$\Updownarrow$$

$$\min \mathbb{E}_{x\sim p(x)}\left[\left[KL(p(z|x)\,||\,q(z)) + \mathbb{E}_{z\sim p(z|x)}[-\log q(x|z)]\right]\right]$$

Derivation for VAE *(Special Case of Information Bottleneck)*

$p(x)$: Evidence probability for the sample $\{x_i\}_1^N$

$q(x)$: Proposed probability to approximate $p(x)$

$q(z)$: Pre-defined prior for the latent vector z

$p(z|x)$: Posterior of z given x: *learned by encoder*

$q(x|z)$: Posterior of x given z: *learned by decoder*

Step 4: Reformulate the goal:

$$\min KL(p(x)||\, q(x))$$

$$\Updownarrow$$

$$\min \mathbb{E}_{x\sim p(x)}\left[\left[KL(p(z|x)\,||\,q(z)) + \mathbb{E}_{z\sim p(z|x)}[-\log q(x|z)]\right]\right]$$

$p(z|x)$: encoder; $q(z)$: pre-defined prior; $q(x|z)$: decoder

Derivation for VAE *(Special Case of Information Bottleneck)*

$p(x)$: Evidence probability for the sample $\{x_i\}_1^N$

$q(x)$: Proposed probability to approximate $p(x)$

$q(z)$: Pre-defined prior for the latent vector z

$p(z|x)$: Posterior of z given x: *learned by encoder*

$q(x|z)$: Posterior of x given z: *learned by decoder*

Step 4: Reformulate the goal:

$$\min KL(p(x)||\, q(x))$$

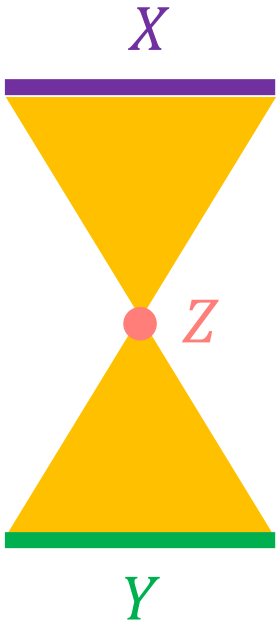
$$\Updownarrow$$

$$\min \mathbb{E}_{x\sim p(x)}\Big[KL(p(z|x)\,||\,q(z)) + \mathbb{E}_{z\sim p(z|x)}[-\log q(x|z)]\Big]$$

$KL(p(z|x)\,||\,q(z))$: *Regularizer*

$\mathbb{E}_{z\sim p(z|x)}[-\log q(x|z)]$: *Reconstruction Loss*

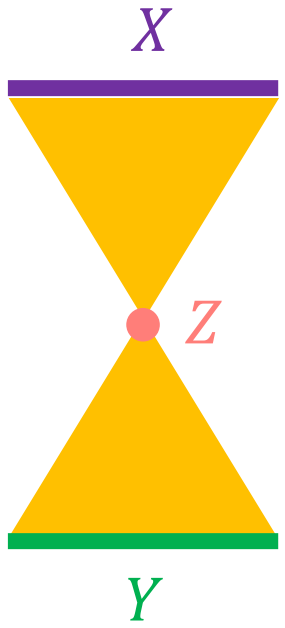
Variational Information Bottleneck (VIB)



$$\max_{\boldsymbol{\theta}} I(Z, Y; \boldsymbol{\theta})$$

$$\text{s.t. } I(X, Z; \boldsymbol{\theta}) \leq I_c$$

Variational Information Bottleneck (VIB)



$$\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Solve for $I(Z, Y)$

$$I(Z, Y) = \int p(y, z) \log \frac{p(y|z)}{p(y)} dydz$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Solve for $I(Z, Y)$

$$I(Z, Y) = \int p(y, z) \log \frac{p(y|z)}{p(y)} dy dz$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Solve for $I(Z, Y)$

$$I(Z,Y) = \int p(y,z) \log \frac{p(y|z)}{p(y)} dydz$$

$$\geq \int p(y,z) \log \frac{q(y|z)}{p(y)} dydz \longrightarrow \text{Decoder!}$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Solve for $I(Z, Y)$

$$
\begin{aligned}
I(Z, Y) &= \int p(y, z) \log \frac{p(y|z)}{p(y)} dydz \\
&\geq \int p(y, z) \log \frac{q(y|z)}{p(y)} dydz \quad \longrightarrow \text{Decoder!} \\
&= \int p(y, z) \log q(y|z) dydz - \int p(y) \log q(y) dy
\end{aligned}
$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Solve for $I(Z, Y)$

$$
\begin{aligned}
I(Z,Y) &= \int p(y,z) \log \frac{p(y|z)}{p(y)} dydz \\
&\geq \int p(y,z) \log \frac{q(y|z)}{p(y)} dydz \quad \longrightarrow \text{Decoder!} \\
&= \int p(y,z) \log q(y|z) dydz - \int p(y) \log q(y) dy \\
&\geq \int p(y,z) \log q(y|z) dydz
\end{aligned}
$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Solve for $I(Z, X)$

$$I(Z, X) = \int p(x, z) \log \frac{p(z|x)}{p(z)} dx dz$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Solve for $I(Z, X)$

$$I(Z, X) = \int p(x, z) \log \frac{p(z|x)}{p(z)} dx dz$$

→ Encoder!

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Solve for $I(Z, X)$

$$I(Z, X) = \int p(x, z) \log \frac{p(z|x)}{p(z)} dx dz$$

→ Encoder!

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Solve for $I(Z, X)$

$$I(Z, X) = \int p(x, z) \log \frac{p(z|x)}{p(z)} dxdz \quad \rightarrow \text{Encoder!}$$

$$\leq \int p(x, z) \log \frac{p(z|x)}{q(z)} dxdz \quad \rightarrow \text{Pre-defined Prior!}$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Solve for $I(Z, X)$

$$I(Z, X) = \int p(x, z) \log \frac{p(z|x)}{p(z)} dx dz \quad \rightarrow \text{Encoder!}$$

$$\leq \int p(x, z) \log \frac{p(z|x)}{q(z)} dx dz \quad \rightarrow \text{Pre-defined Prior!}$$

$$= \int p(x) p(z|x) \log \frac{p(z|x)}{q(z)} dx dz$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Put together…

$$I(Z,Y) - \beta I(Z,X) \geq \int p(x)p(y|x)y(z|x) \log q(y|z) dx dy dz - \beta \int p(x)p(z|x) \log \frac{p(z|x)}{q(z)} dx dz$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Put together…

$$I(Z,Y) - \beta I(Z,X) \geq \int p(x)p(y|x)y(z|x) \log q(y|z) dx dy dz - \beta \int p(x)p(z|x) \log \frac{p(z|x)}{q(z)} dx dz$$

$$= \mathbb{E}_{x \sim p(x)} \left[\mathbb{E}_{z \sim p(z|x)} [-\log q(y|z)] + \beta \cdot KL(p(z|x) \| q(z)) \right]$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Put together…

$$I(Z,Y) - \beta I(Z,X) \geq \int p(x)p(y|x)y(z|x) \log q(y|z) dx dy dz - \beta \int p(x)p(z|x) \log \frac{p(z|x)}{q(z)} dx dz$$

$$= \mathbb{E}_{x \sim p(x)} \left[\mathbb{E}_{z \sim p(z|x)} [-\log q(y|z)] + \beta \cdot KL(p(z|x) \| q(z)) \right]$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Put together…

$$I(Z,Y) - \beta I(Z,X) \geq \int p(x)p(y|x)y(z|x) \log q(y|z) dx dy dz - \beta \int p(x)p(z|x) \log \frac{p(z|x)}{q(z)} dx dz$$

$$= \mathbb{E}_{x \sim p(x)} \left[\mathbb{E}_{z \sim p(z|x)} [-\log q(y|z)] + \beta \cdot KL(p(z|x) \| q(z)) \right]$$

$$= \mathbb{E}_{x \sim p(x)} \left[\beta \cdot KL(p(z|x) \| q(z)) + \mathbb{E}_{z \sim p(z|x)} [-\log q(y|z)] \right]$$

Solve for $\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$

Put together…

$$I(Z,Y) - \beta I(Z,X) \geq \int p(x)p(y|x)y(z|x) \log q(y|z) dxdydz - \beta \int p(x)p(z|x) \log \frac{p(z|x)}{q(z)} dxdz$$

$$= \mathbb{E}_{x \sim p(x)} \left[\mathbb{E}_{z \sim p(z|x)} [-\log q(y|z)] + \beta \cdot KL(p(z|x) \| q(z)) \right]$$

$$= \mathbb{E}_{x \sim p(x)} \left[\beta \cdot KL(p(z|x) \| q(z)) + \mathbb{E}_{z \sim p(z|x)} [-\log q(y|z)] \right]$$

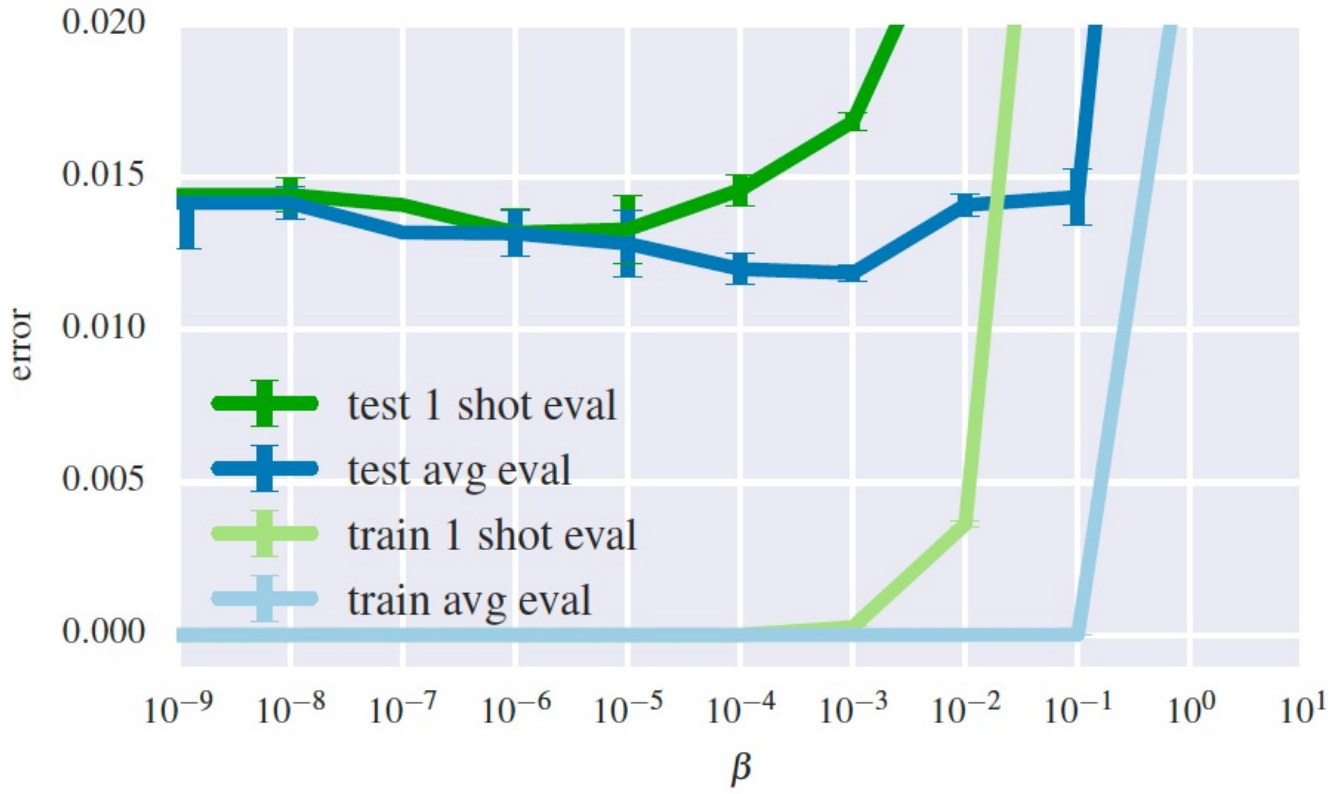
VAE as a special class for Information Bottleneck

$$\min \mathbb{E}_{x \sim p(x)} \left[KL(p(z|x) \,||\, q(z)) + \mathbb{E}_{z \sim p(z|x)} [-\log q(x|z)] \right]$$

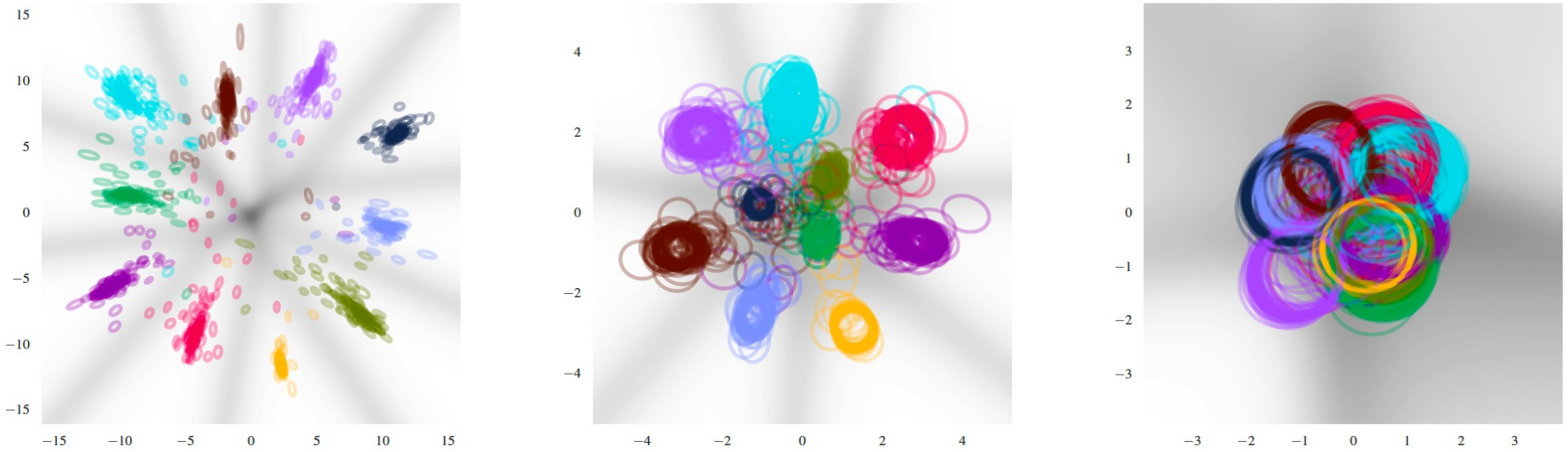
Performance on MNIST

Model	error
Baseline	1.38%
Dropout	1.34%
Dropout (Pereyra et al., 2016)	1.40%
Confidence Penalty	1.36%
Confidence Penalty (Pereyra et al., 2016)	1.17%
Label Smoothing	1.40%
Label Smoothing (Pereyra et al., 2016)	1.23%
VIB ($\beta = 10^{-3}$)	**1.13%**

Effect of β on Performance

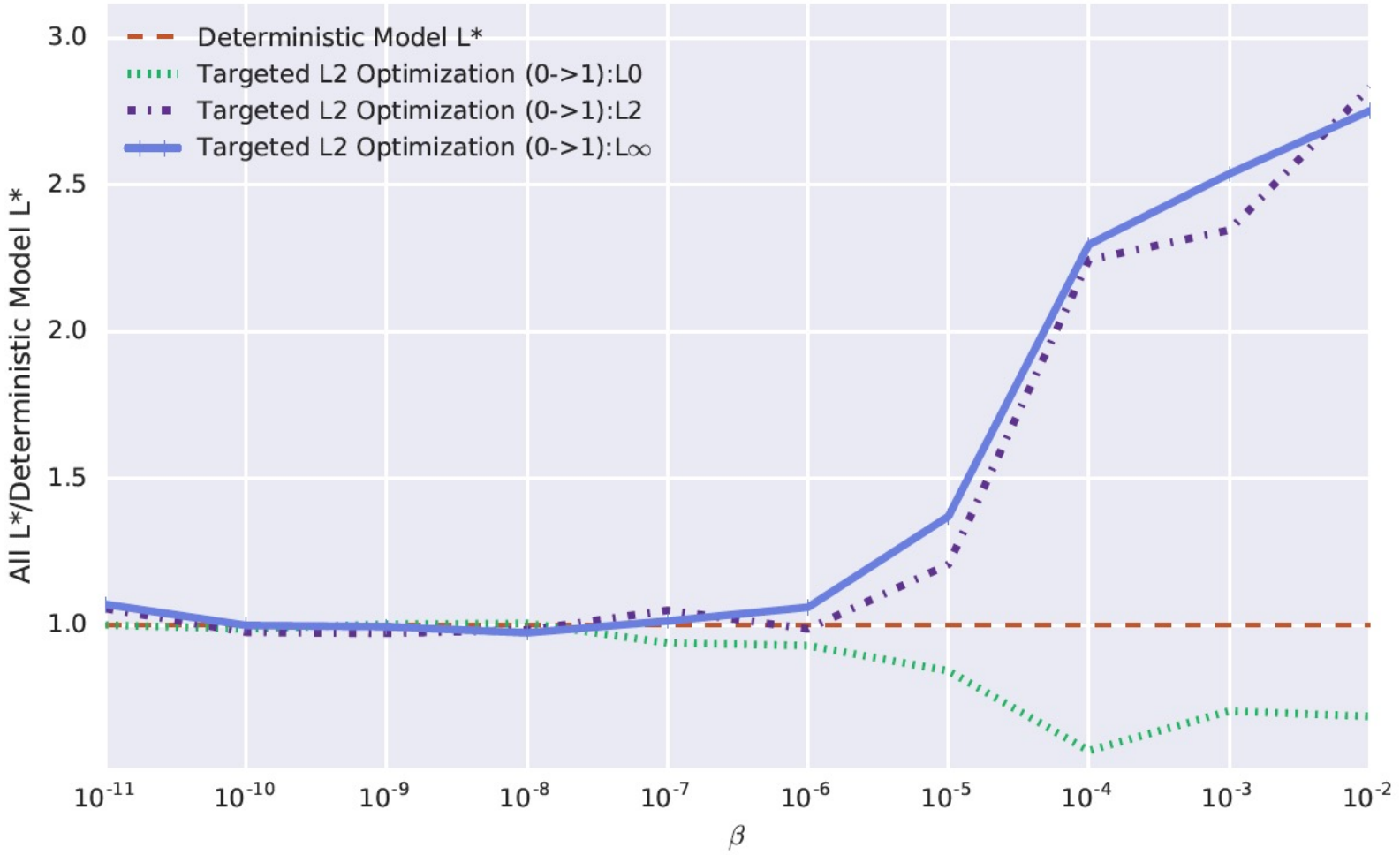


Effect of β on Representation Learning

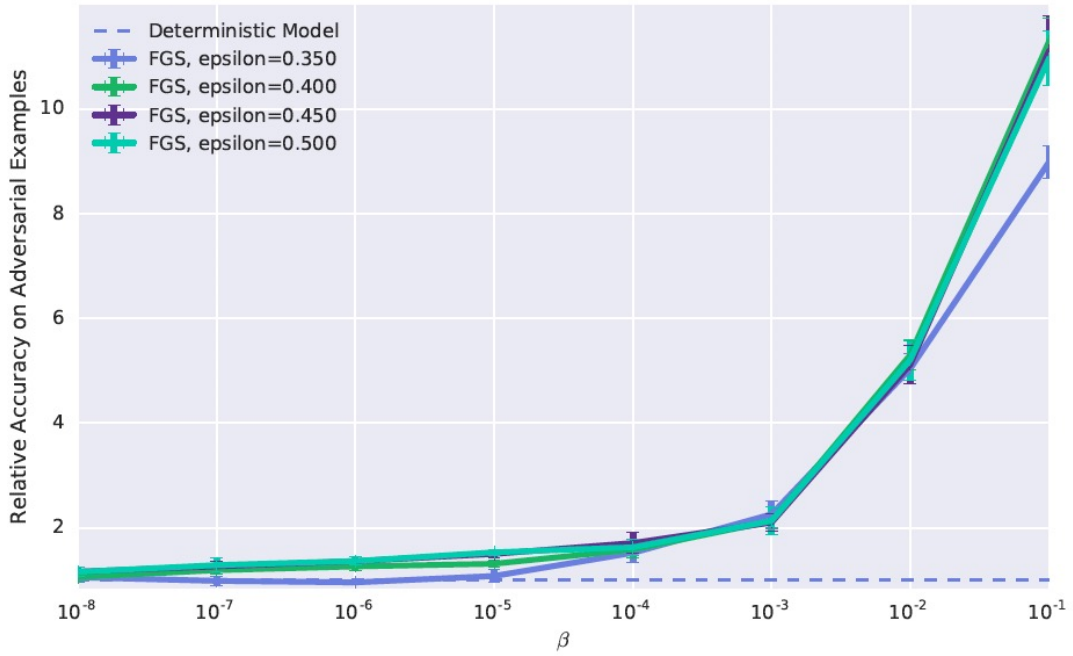


(a) $\beta = 10^{-3}$, $\text{err}_{\text{mc}} = 3.18\%$, $\text{err}_1 = 3.24\%$ (b) $\beta = 10^{-1}$, $\text{err}_{\text{mc}} = 3.44\%$, $\text{err}_1 = 4.32\%$ (c) $\beta = 10^{0}$, $\text{err}_{\text{mc}} = 33.82\%$, $\text{err}_1 = 62.81\%$.

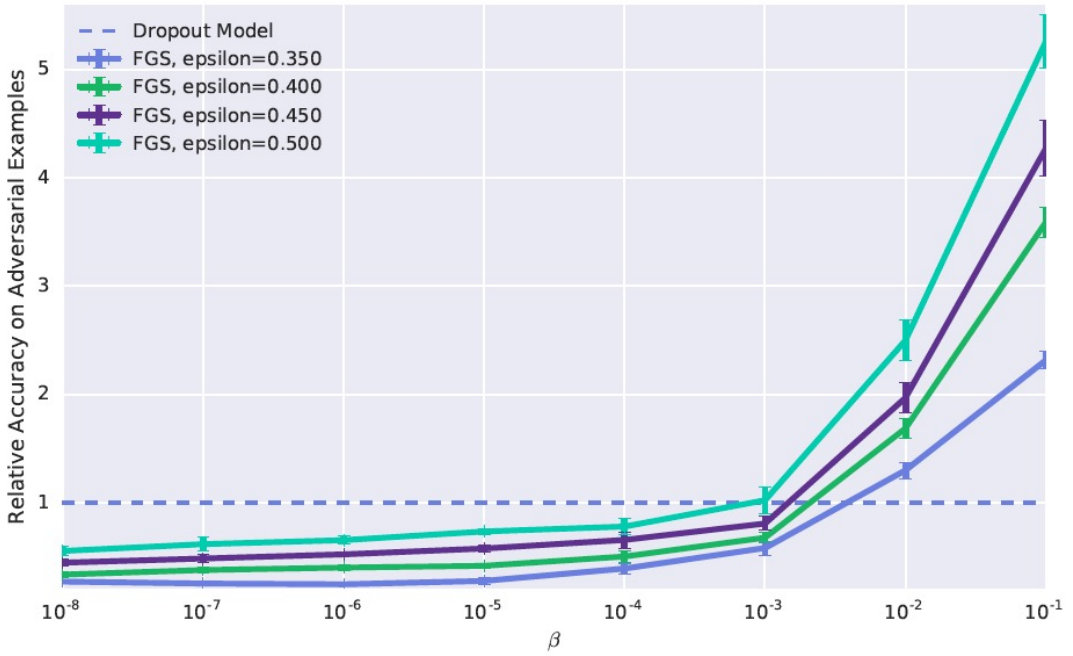
Adversarial Robustness



Effect of β on Relative Accuracy on Adversarial Examples (FGS)



(a)

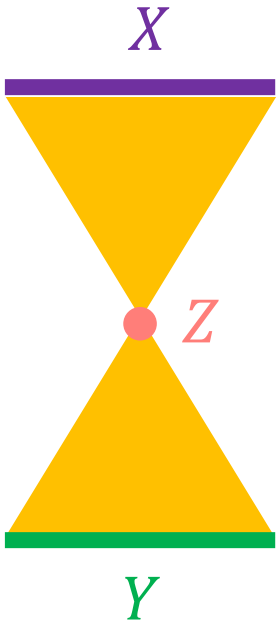


(b)

Adversarial Robustness

Metric	Determ	IRv2	VIB(0.01)
Sucessful target	1.0	1.0	**0.567**
L_2	6.45	14.43	**43.27**
L_∞	0.18	0.44	**0.92**

Conclusions



$$\max_{\theta} I(Z, Y; \theta) - \beta I(Z, X; \theta)$$