Lecture 10: Domain Adaptation and Generalization

Shujian Yu

Deep Learning 2023

dlvu.github.io

VU VRIJE UNIVERSITEIT AMSTERDAM

THE PLAN

part 1: review (distance measures)

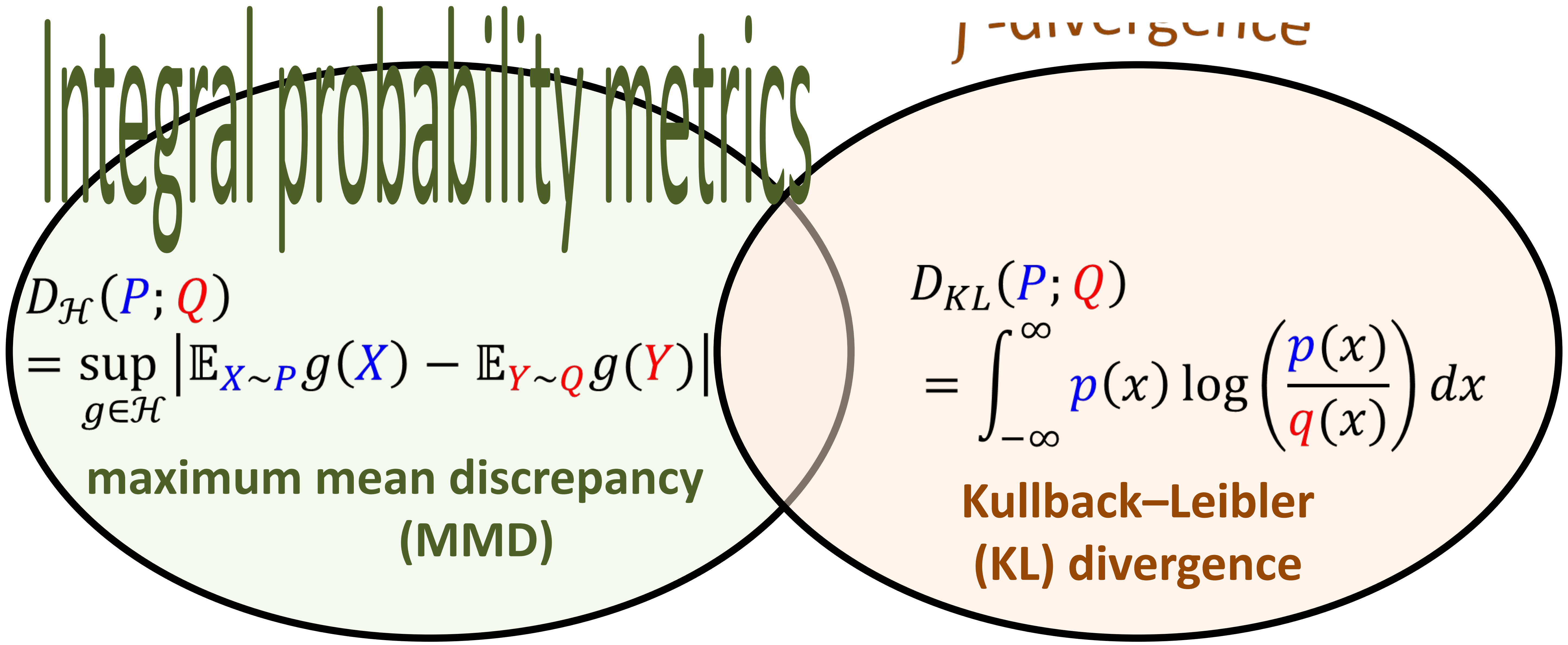
part 2: problem of domain adaptation

part 3: domain adaptation and generalization error bound

part 4: compression and generalization

PART ONE: REVIEW

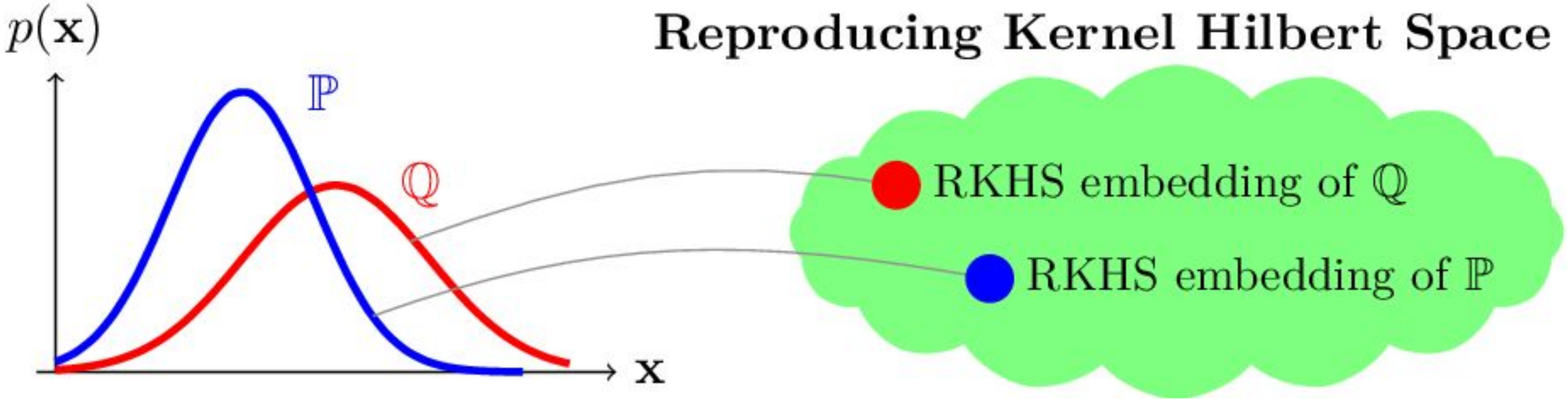
How to measure the distance/divergence?



Gretton, Arthur, et al. "A kernel two-sample test." *The Journal of Machine Learning Research* 13.1 (2012): 723-773. https://www.jmlr.org/papers/volume13/gretton12a/gretton12a.pdf

MAXIMUM MEAN DISCREPANCY

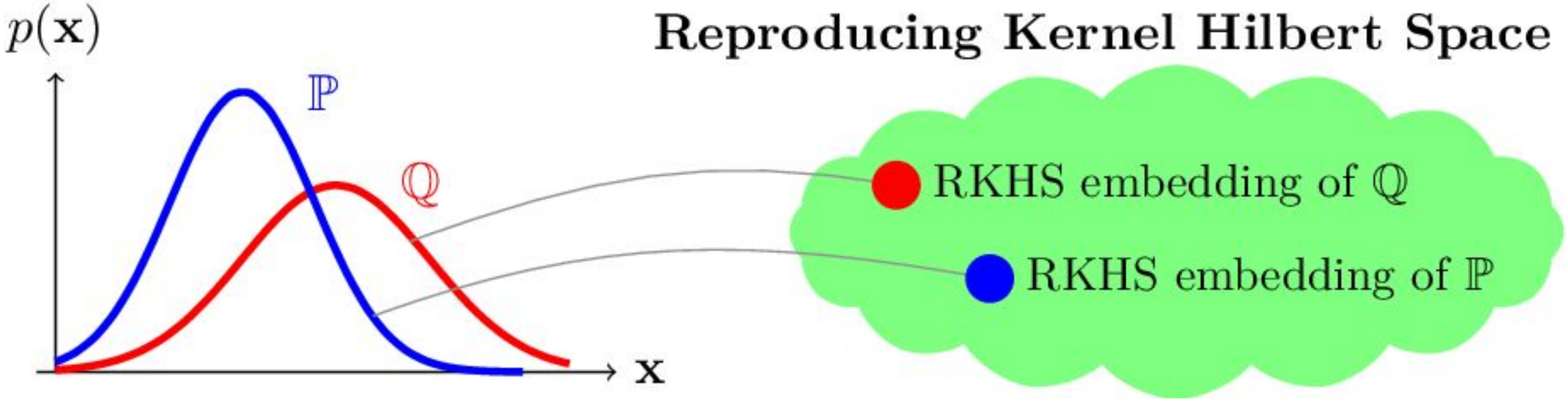
For a feature map $\varphi: \mathcal{X} \rightarrow \mathcal{H}$, representing distances between distributions as distances between mean embeddings of features



$$\mathrm{MMD}^2(P; Q) = \left\| \mathbb{E}_{X \sim P} \varphi(X) - \mathbb{E}_{Y \sim Q} \varphi(Y) \right\|_{\mathcal{H}}^2$$

Muandet, Krikamol, et al. "Kernel mean embedding of distributions: A review and beyond." *Foundations and Trends® in Machine Learning* 10.1-2 (2017): 1-141. https://www.nowpublishers.com/article/Details/MAL-060

For a feature map $\varphi: \mathcal{X} \to \mathcal{H}$, representing distances between distributions as distances between mean embeddings of features



$$\begin{aligned}\mathrm{MMD}^2(P;Q) &= \left\|\mathbb{E}_{X\sim P}\varphi(X) - \mathbb{E}_{Y\sim Q}\varphi(Y)\right\|_{\mathcal{H}}^2 \\ &= \left\langle \mathbb{E}_{X\sim P}\varphi(X), \mathbb{E}_{X'\sim P}\varphi(X')\right\rangle_{\mathcal{H}} + \left\langle \mathbb{E}_{Y\sim Q}\varphi(Y), \mathbb{E}_{Y'\sim Q}\varphi(Y')\right\rangle_{\mathcal{H}} \\ &\qquad -2\left\langle \mathbb{E}_{X\sim P}\varphi(X), \mathbb{E}_{Y\sim Q}\varphi(Y)\right\rangle_{\mathcal{H}} \\ &= \mathbb{E}_{X,X'\sim P}\,\kappa(X,X') + \mathbb{E}_{Y,Y'\sim Q}\kappa(Y,Y') - 2\mathbb{E}_{X\sim P,Y\sim Q}\kappa(X,Y)\end{aligned}$$

$$(x-y)^2 = x^Tx + y^Ty - 2x^Ty$$

The kernel trick: $\kappa(x,y) = \langle\varphi(x),\varphi(y)\rangle_{\mathcal{H}}$

MAXIMUM MEAN DISCREPANCY

For a feature map $\varphi\colon \mathcal{X} \to \mathcal{H}$, representing distances between distributions as distances between mean embeddings of features

$$\mathrm{MMD}^2(P;Q) = \mathbb{E}_{X,X'\sim P}\kappa(X,X') + \mathbb{E}_{Y,Y'\sim Q}\kappa(Y,Y') - 2\mathbb{E}_{X\sim P,Y\sim Q}\kappa(X,Y)$$

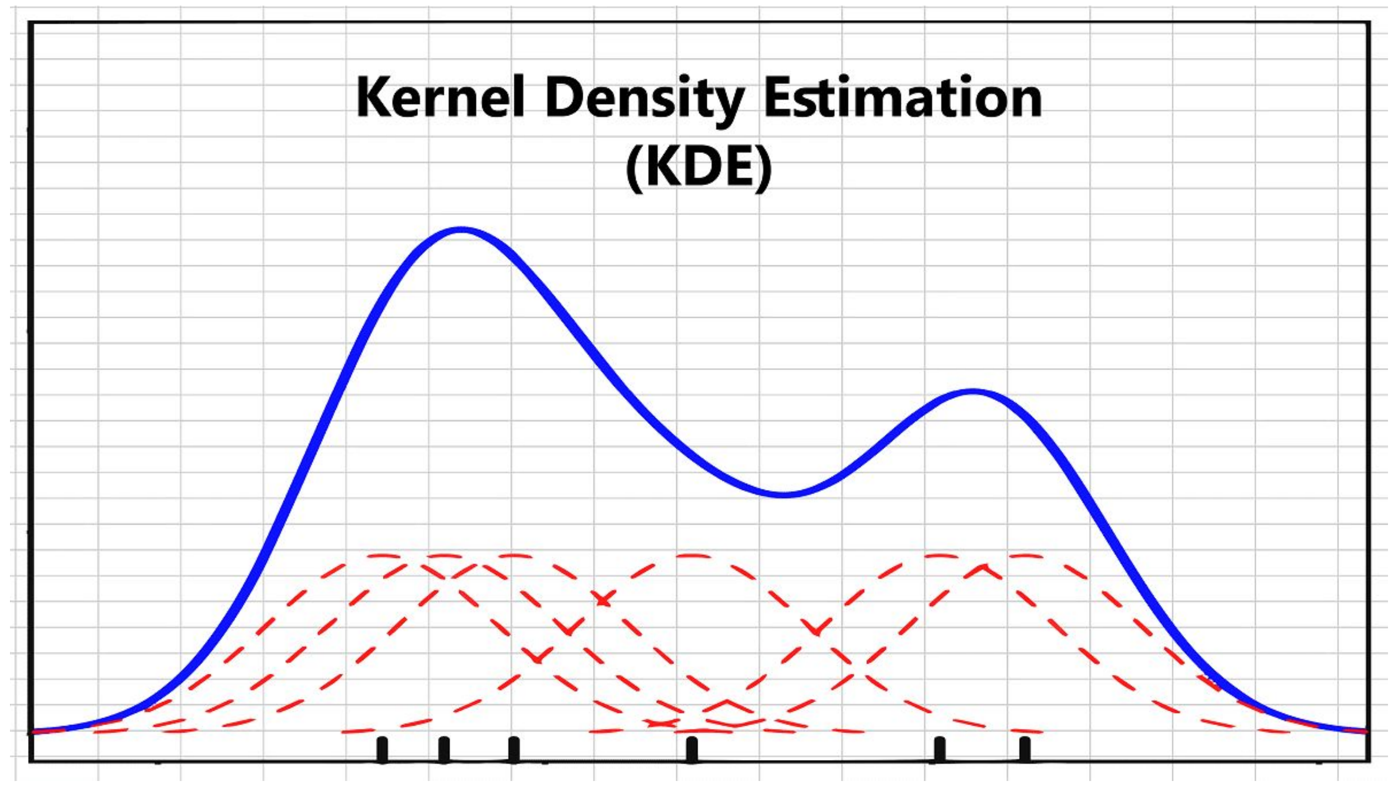
$$\widehat{\mathrm{MMD}}^2(P;Q) = \underbrace{\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_\sigma(x_i - x_j)}_{\text{within distribution similarity}} + \underbrace{\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} G_\sigma(y_i - y_j)}_{\text{within distribution similarity}} - \underbrace{\frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} G_\sigma(x_i - y_j)}_{\text{cross-distribution similarity}}$$

Gretton, Arthur, et al. "A kernel two-sample test." *The Journal of Machine Learning Research* 13.1 (2012): 723-773.

Euclidean distance

$$D_{\mathrm{ED}} = \int (p-q)^2 d\mu$$

$$D_{\mathrm{ED}} = \int p^2 d\mu - 2\int pq\, d\mu + \int q^2 d\mu$$



$$\hat{p}_s(\mathbf{y}) = \frac{1}{M}\sum_{i=1}^{M} G_\sigma(\mathbf{y}-\mathbf{y}_i^s)$$

$$\begin{aligned}
\int \hat{p}_s^2(\mathbf{y})dy &= \int \left(\frac{1}{M}\sum_{i=1}^{M} G_\sigma(\mathbf{y}-\mathbf{y}_i^s)\right)^2 dy \\
&= \frac{1}{M^2}\int \left(\sum_{i=1}^{M}\sum_{j=1}^{M} G_\sigma(\mathbf{y}-\mathbf{y}_j^s)\cdot G_\sigma(\mathbf{y}-\mathbf{y}_i^s)\right) dy \\
&= \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\int G_\sigma(\mathbf{y}-\mathbf{y}_j^s)\cdot G_\sigma(\mathbf{y}-\mathbf{y}_i^s)dy \\
&= \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} G_{\sqrt{2}\sigma}(\mathbf{y}_j^s-\mathbf{y}_i^s).
\end{aligned}$$

Principe, Jose C., et al. "Learning from examples with information theoretic criteria." *Journal of VLSI signal processing systems for signal, image and video technology* 26 (2000): 61-77.

Euclidean distance

$$D_{\mathrm{ED}} = \int (p-q)^2 d\mu$$

$$\begin{aligned} D_{\mathrm{ED}} &= \int p^2 d\mu - 2\int pq d\mu + \int q^2 d\mu \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_i - x_j) - \frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} G_{\sigma\sqrt{2}}(x_i - y_j) \\ &+ \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} G_{\sigma\sqrt{2}}(y_i - y_j) \end{aligned}$$

Exactly the expression of the famed maximum mean discrepancy (MMD)!

KULLBACK–LEIBLER DIVERGENCE

Kullback-Leibler (KL) Divergence measures the “distance” between probability density functions (pdfs)

- relative entropy
- Cross entropy ($D_{\mathrm{KL}}(p;q) = \mathrm{CE}(p,q) - H(p)$).
- information for discrimination

$$D_{\mathrm{KL}}(p;q) = \int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

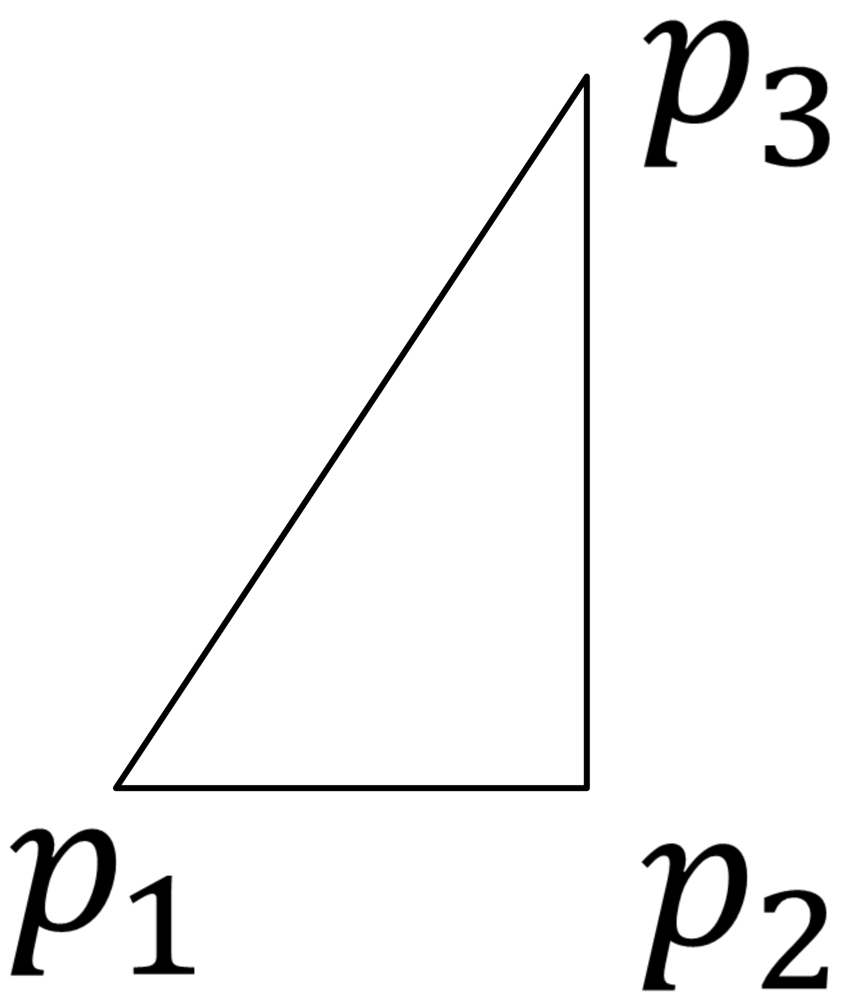
KULLBACK–LEIBLER DIVERGENCE

distance definition

- non-negative
- null only if pdfs are equal
- symmetric
- triangle inequality

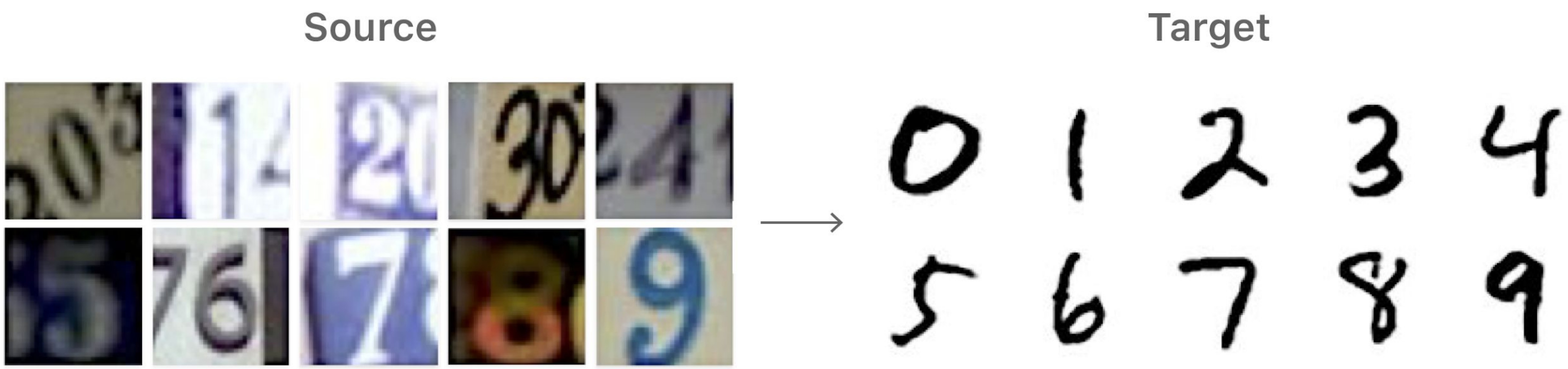
in reality

- $D_{\mathrm{KL}}(p;q) \geq 0$
- $D_{\mathrm{KL}}(p;q) = 0$, iff $p = q$
- $D_{\mathrm{KL}}(p;q) \neq D_{\mathrm{KL}}(q;p)$
- $D_{\mathrm{KL}}(p_1;p_2) + D_{\mathrm{KL}}(p_2;p_3)$ NOT $\geq D_{\mathrm{KL}}\ (p_1;p_3)$



PART TWO: PROBLEM OF DOMAIN ADAPTATION

PROBLEM OF DOMAIN ADAPTATION



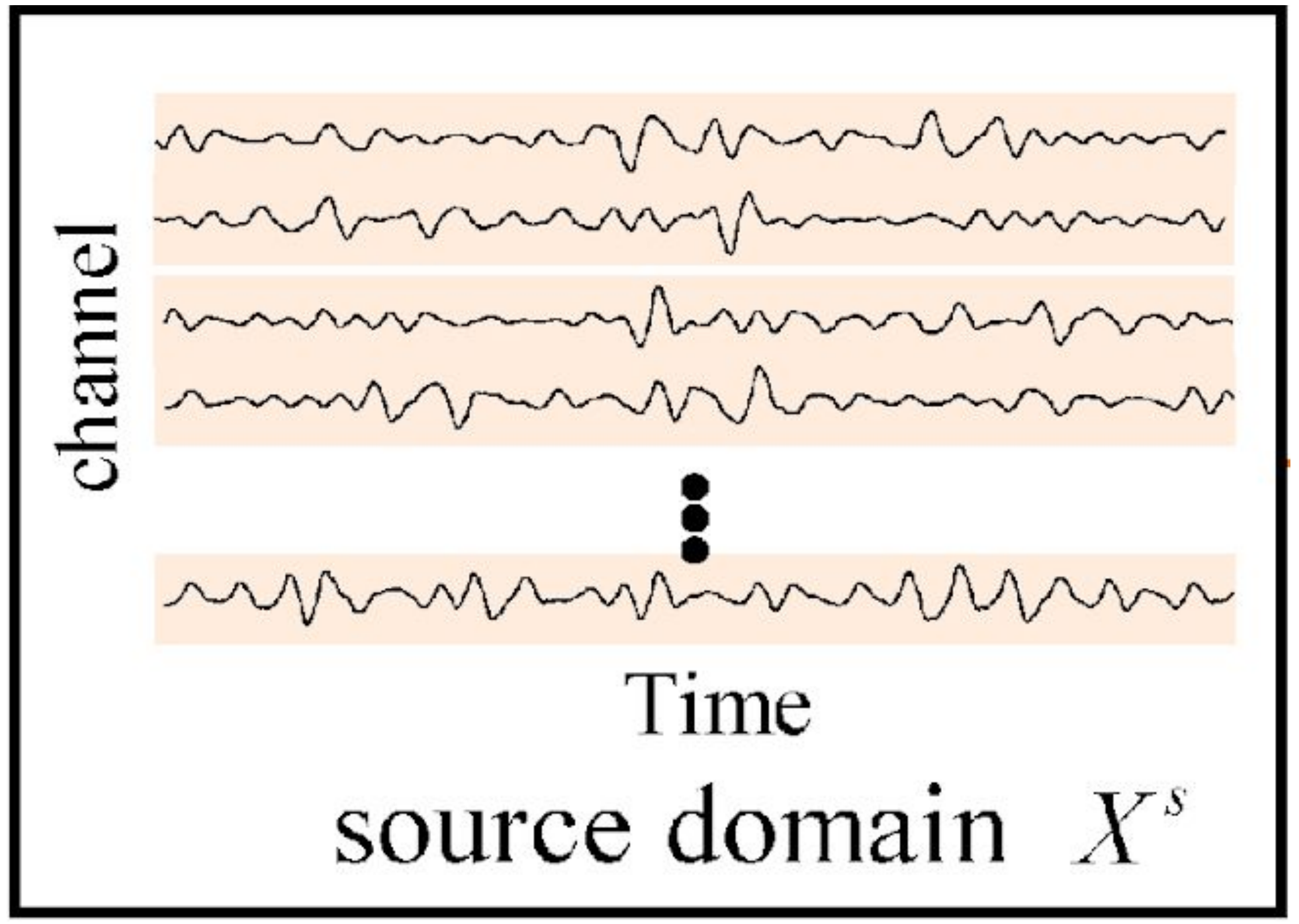
Training data: SVHN dataset

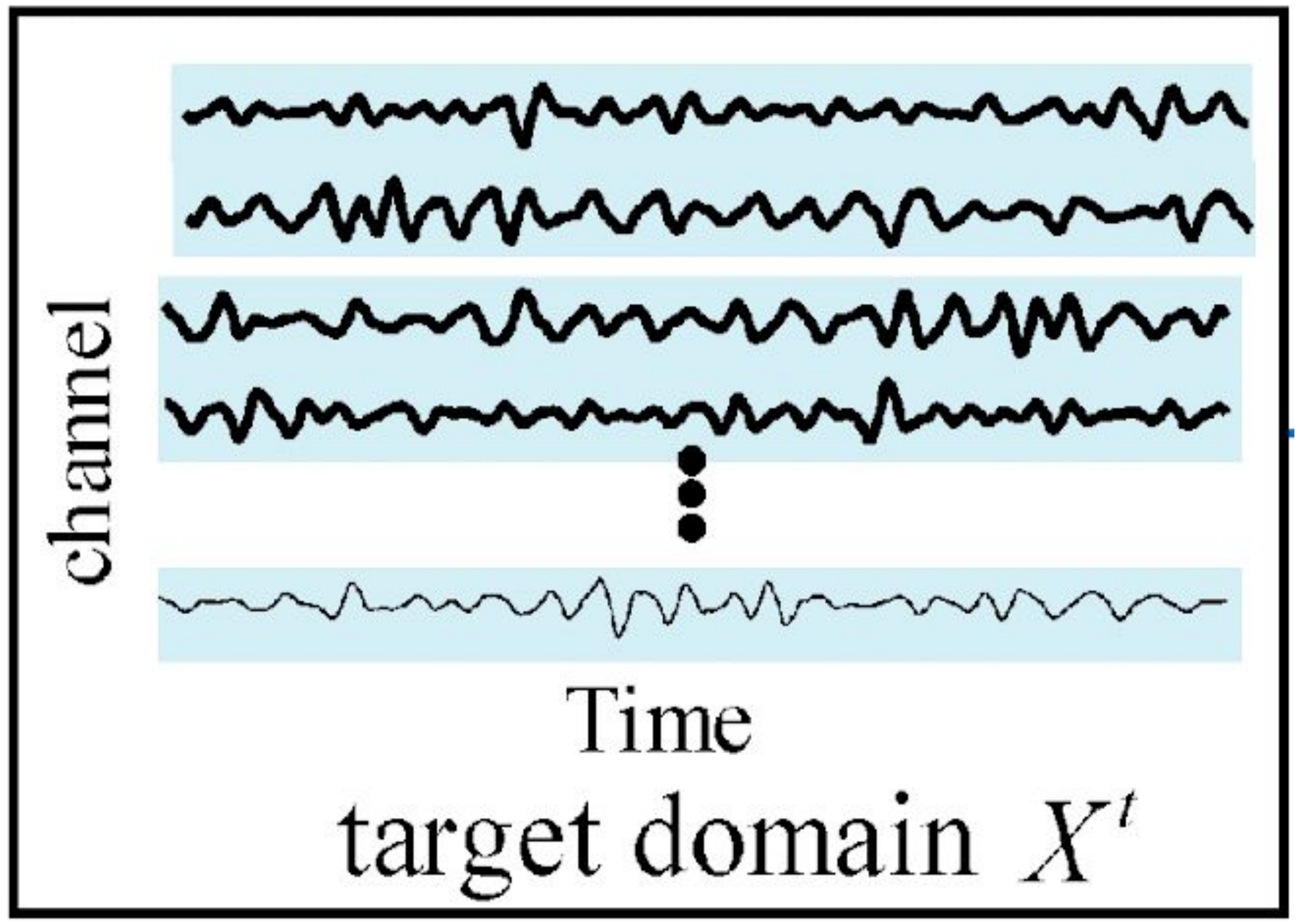
Test data: MNIST dataset

https://machinelearning.apple.com/research/bridging-the-domain-gap-for-neural-models

Netzer, Yuval, et al. "Reading digits in natural images with unsupervised feature learning." (2011).
LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

PROBLEM OF DOMAIN ADAPTATION





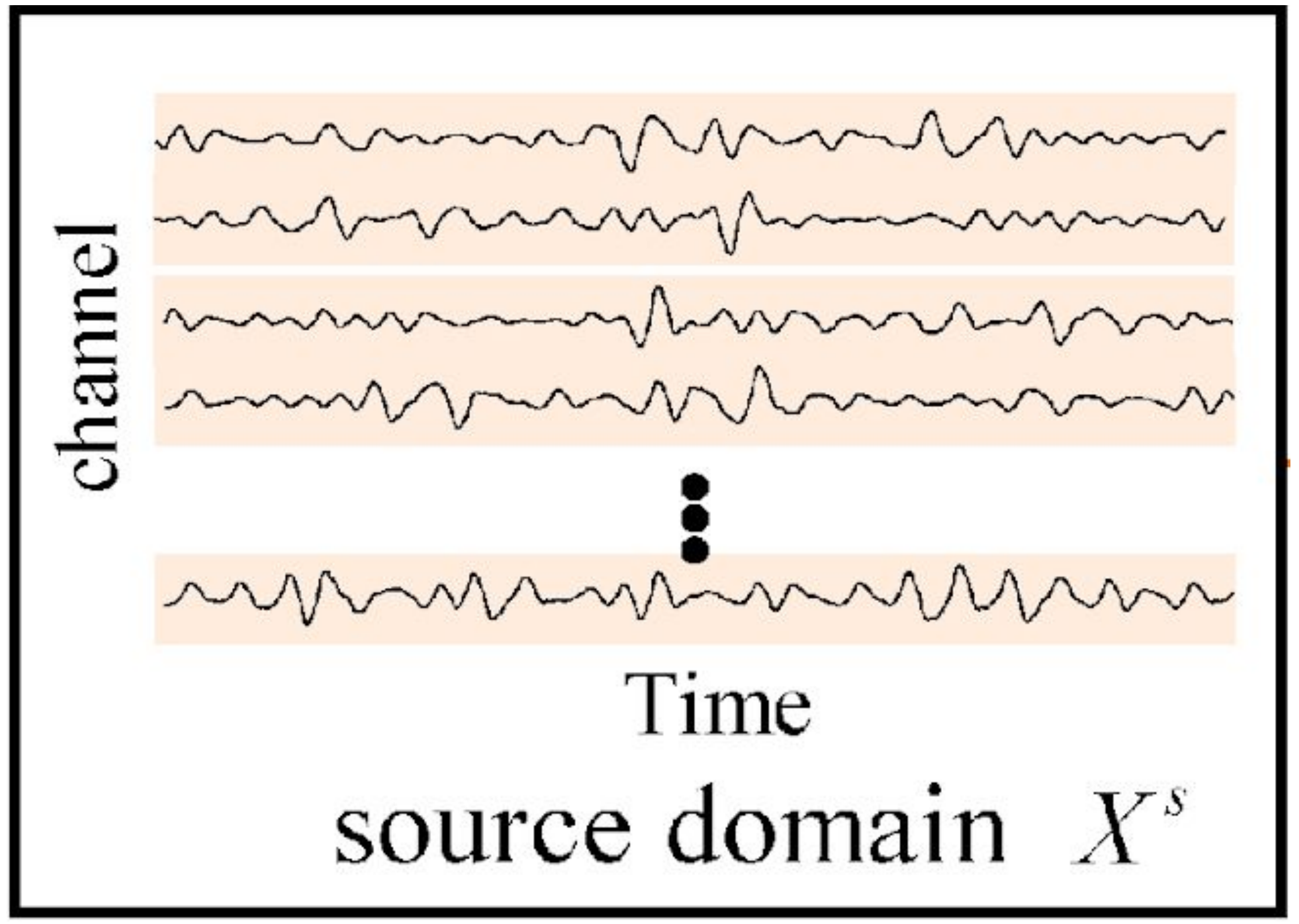
Training data: EEG signals from hospital A

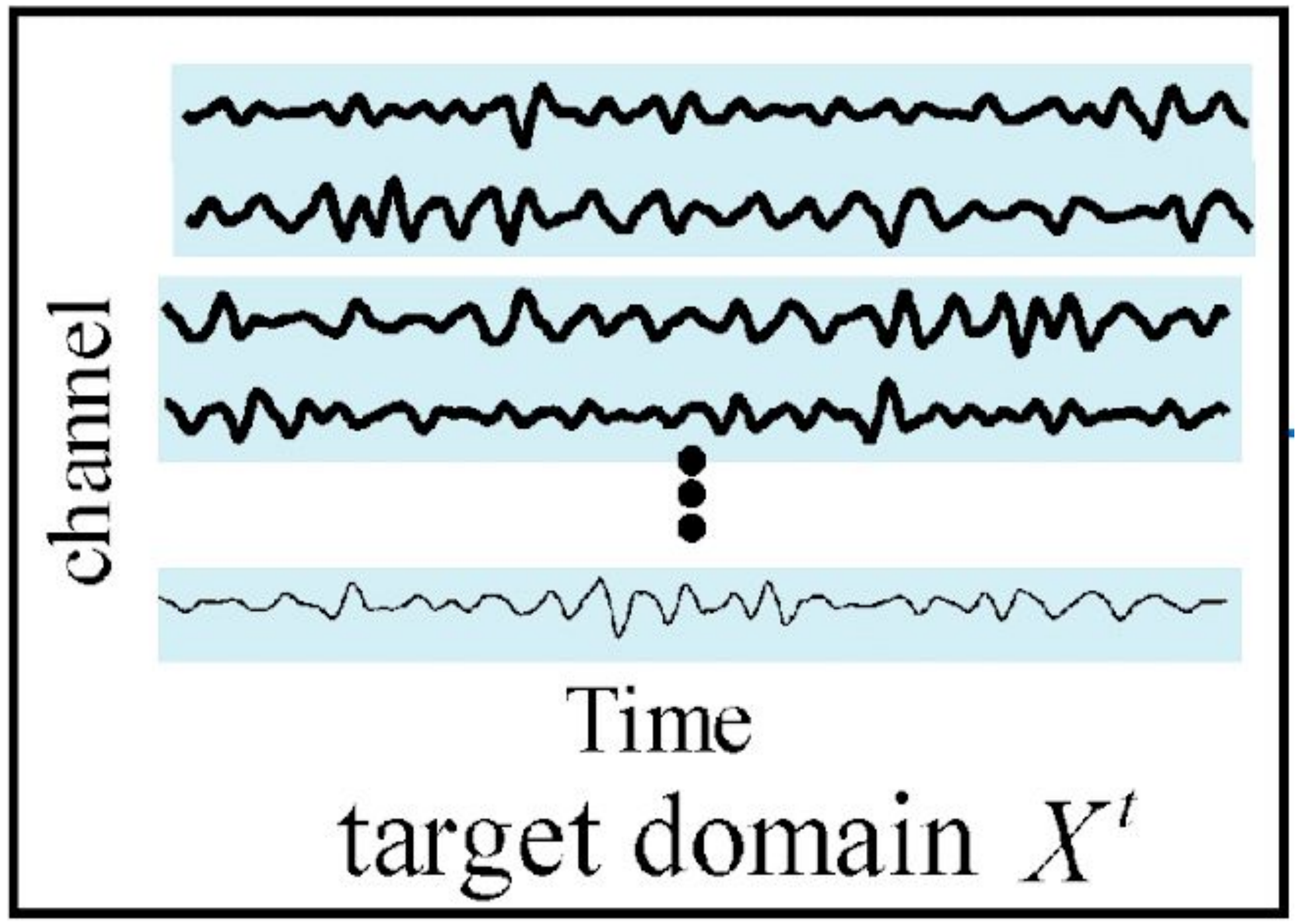
Test data: EEG signals from hospital B

Training data: EEG signals collected in 2020

Test data: EEG signals collected in 2023

Tang, Xingliang, and Xianrui Zhang. "Conditional adversarial domain adaptation neural network for motor imagery EEG decoding." Entropy 22.1 (2020): 96. https://www.mdpi.com/1099-4300/22/1/96





Training data: EEG signals from hospital A

Test data: EEG signals from hospital B

Training data: EEG signals collected in 2020

Test data: EEG signals collected in 2023

DISTRIBUTIONAL SHIFT

$$p_s(\boldsymbol{x}, y) \neq p_t(\boldsymbol{x}, y)$$

$$p_s(\boldsymbol{x}, y) = p_s(\boldsymbol{x})p_s(y|\boldsymbol{x})$$

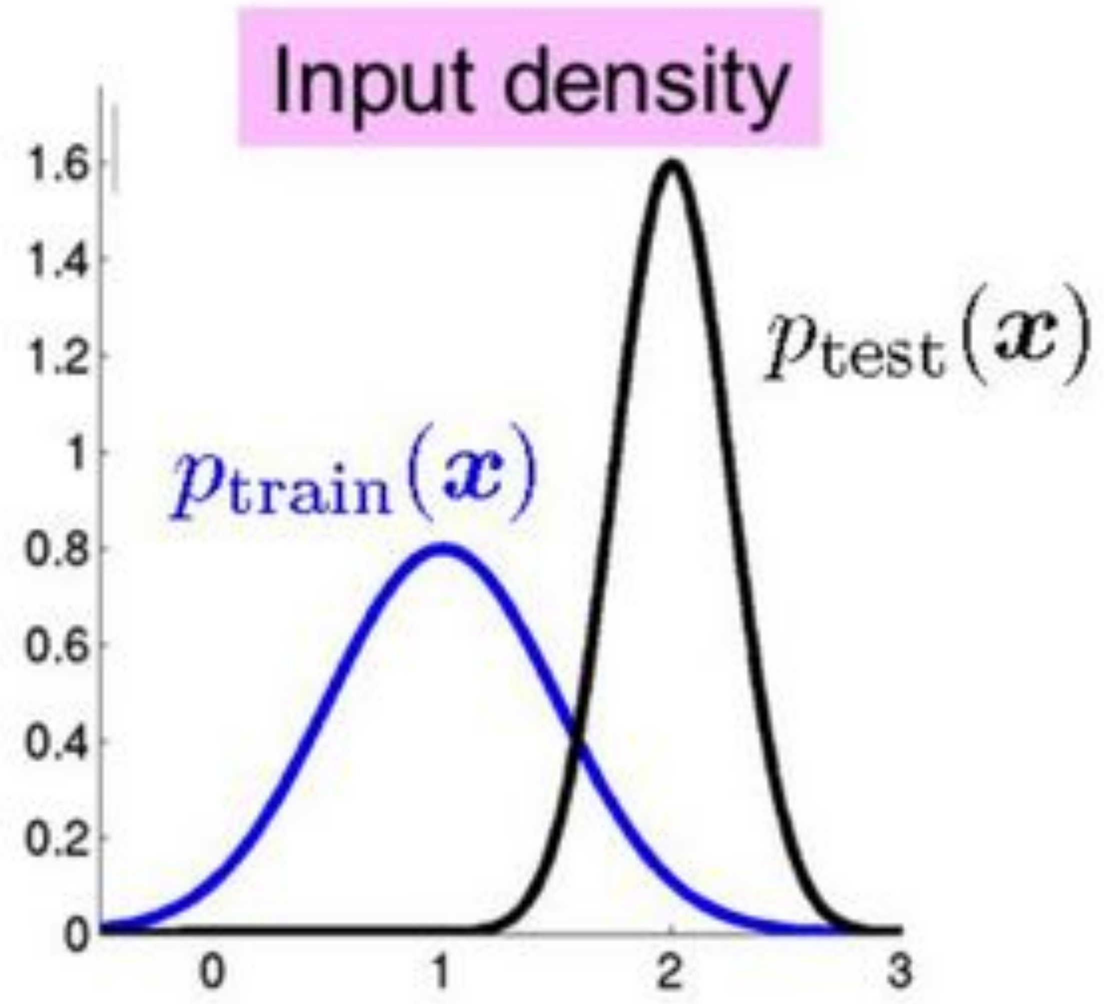
$$p_t(\boldsymbol{x}, y) = p_t(\boldsymbol{x})p_t(y|\boldsymbol{x})$$

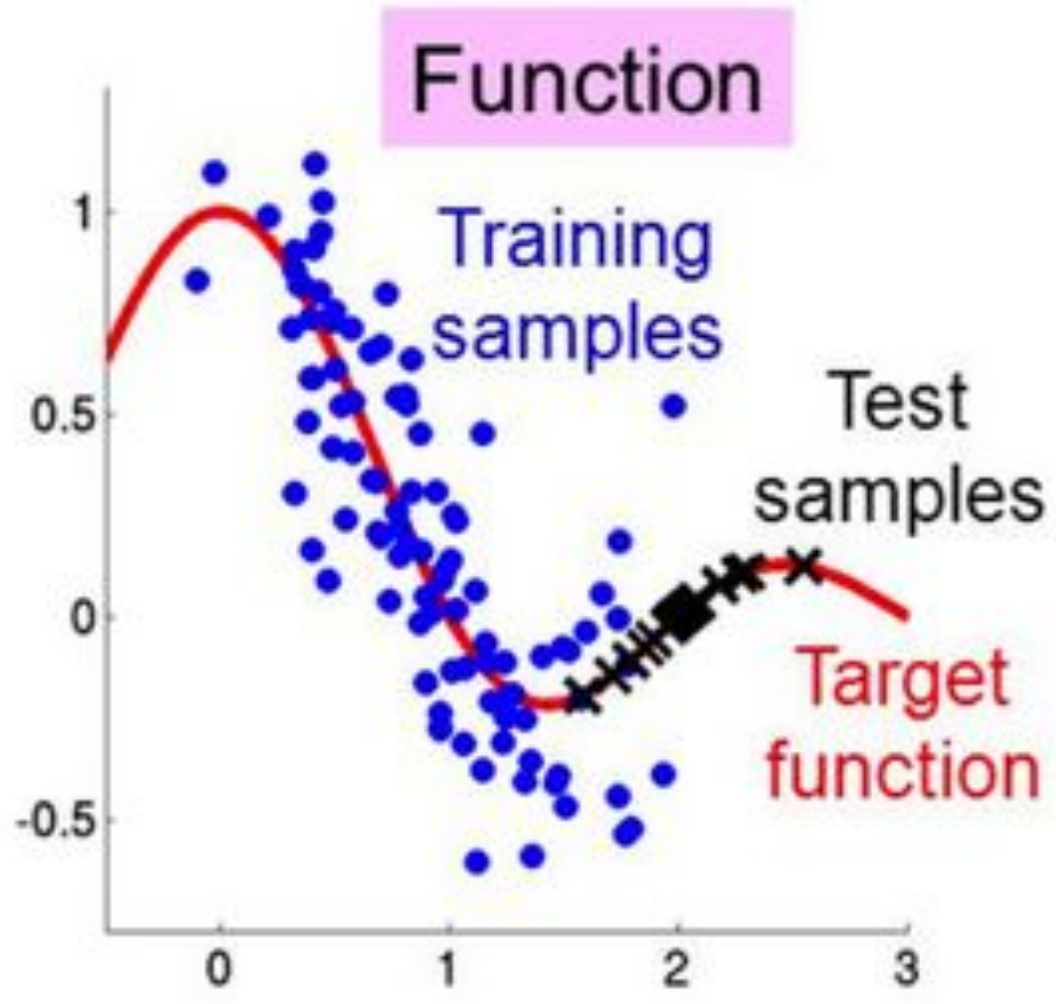
distance between source and target **feature**

$$p_s(\boldsymbol{x}, y) \neq p_t(\boldsymbol{x}, y)$$

$$p_s(\boldsymbol{x}, y) = p_s(\boldsymbol{x}) p_s(y|\boldsymbol{x})$$

$$p_t(\boldsymbol{x}, y) = p_t(\boldsymbol{x}) p_t(y|\boldsymbol{x})$$



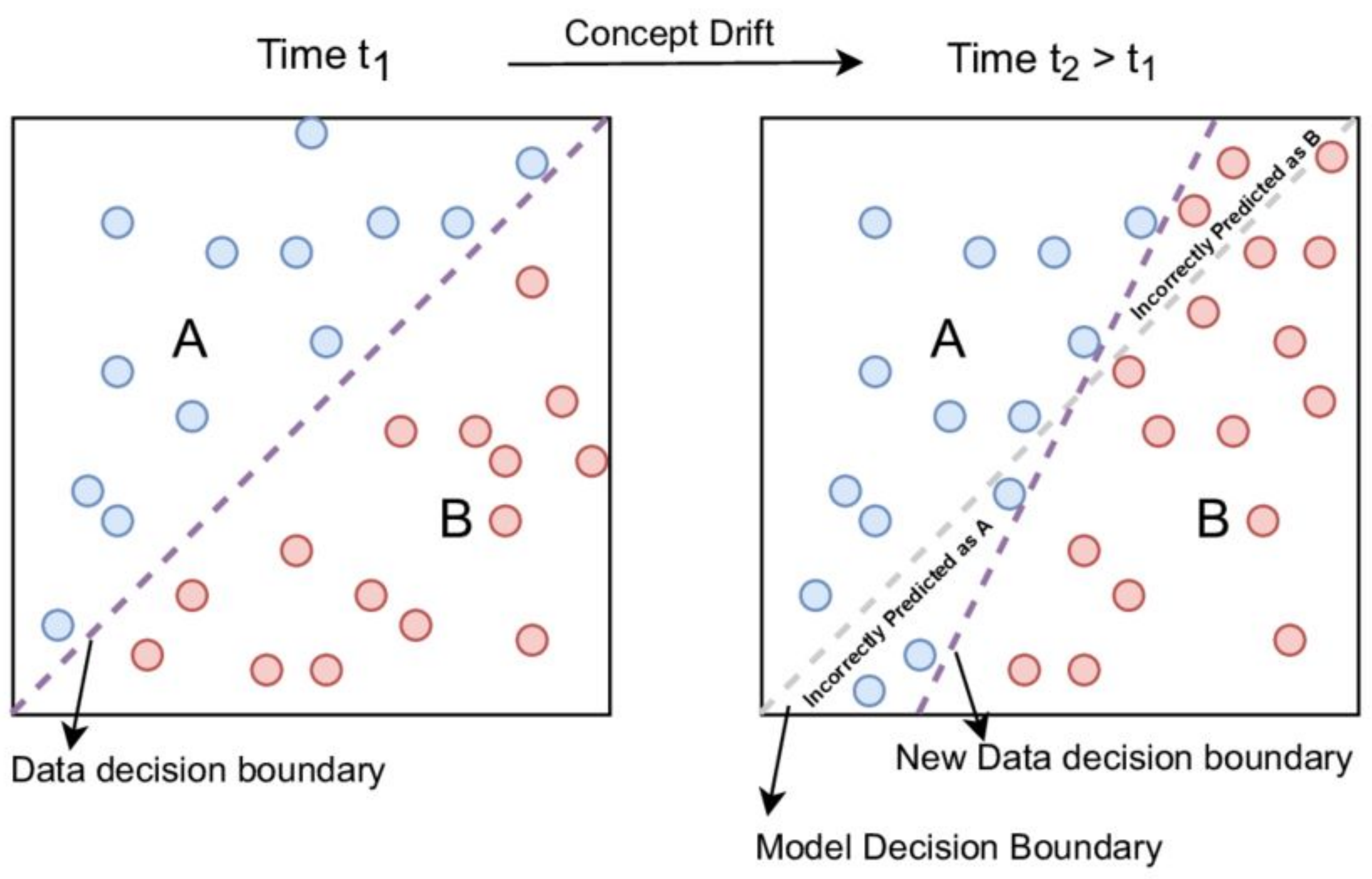


distance between source and target **labeling function**

$$p_s(\boldsymbol{x}, y) \neq p_t(\boldsymbol{x}, y)$$

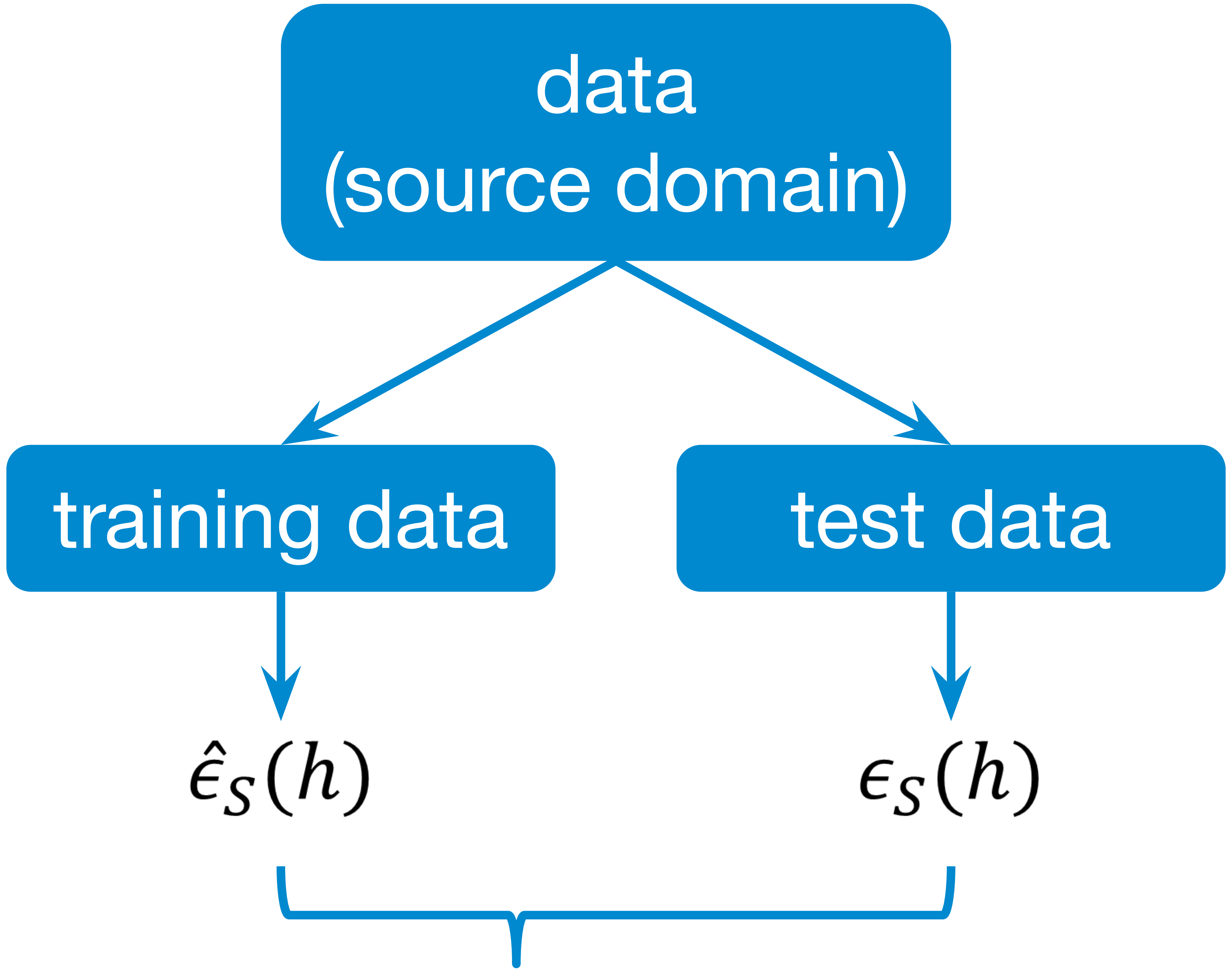
$$p_s(\boldsymbol{x}, y) = p_s(\boldsymbol{x})p_s(y|\boldsymbol{x})$$

$$p_t(\boldsymbol{x}, y) = p_t(\boldsymbol{x})p_t(y|\boldsymbol{x})$$



PART THREE: GENERALIZATION BOUND OF DOMAIN ADAPTATION

How to bound risks in target domain? How to design practical algorithms?



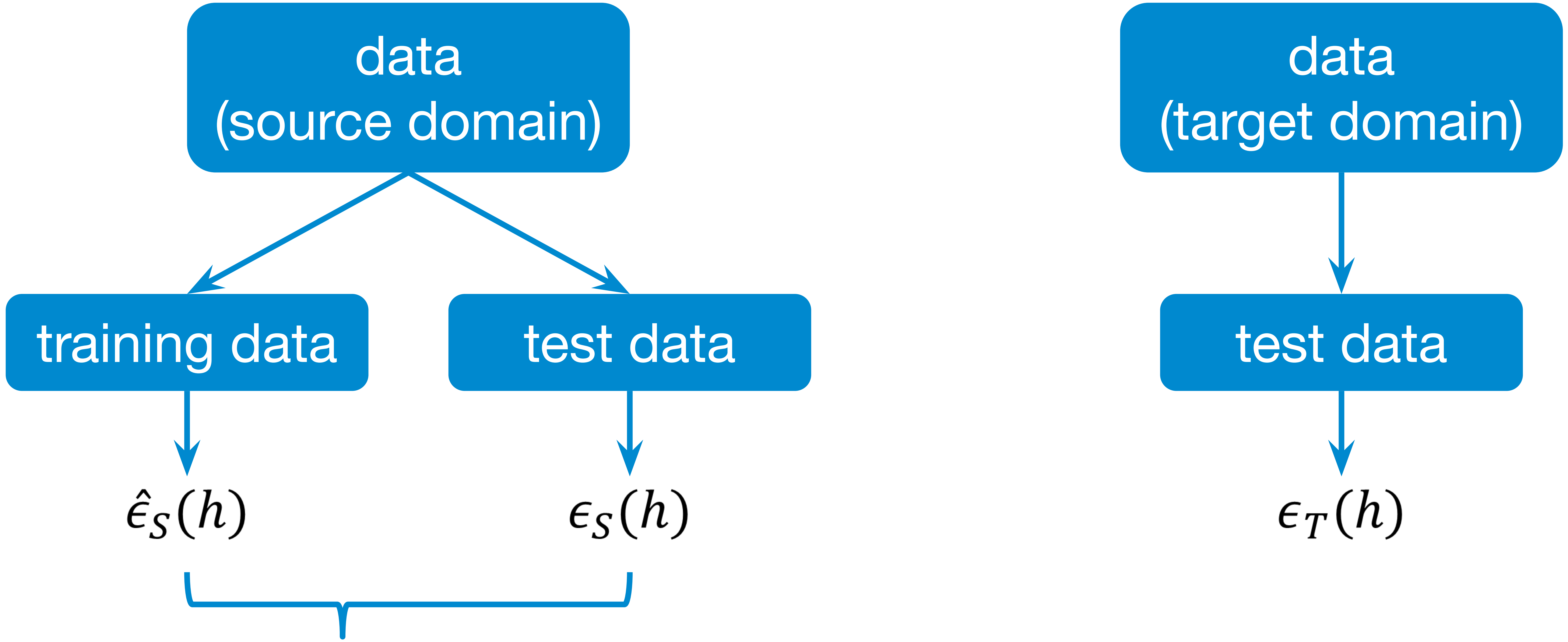
$$\epsilon_S(h) \leq \hat{\epsilon}_S(h) + \sqrt{\frac{8}{n}\log\left(\frac{4(2n)^d}{\delta}\right)}$$

n: the number of training instances

d: VC-dimension (model complexity)

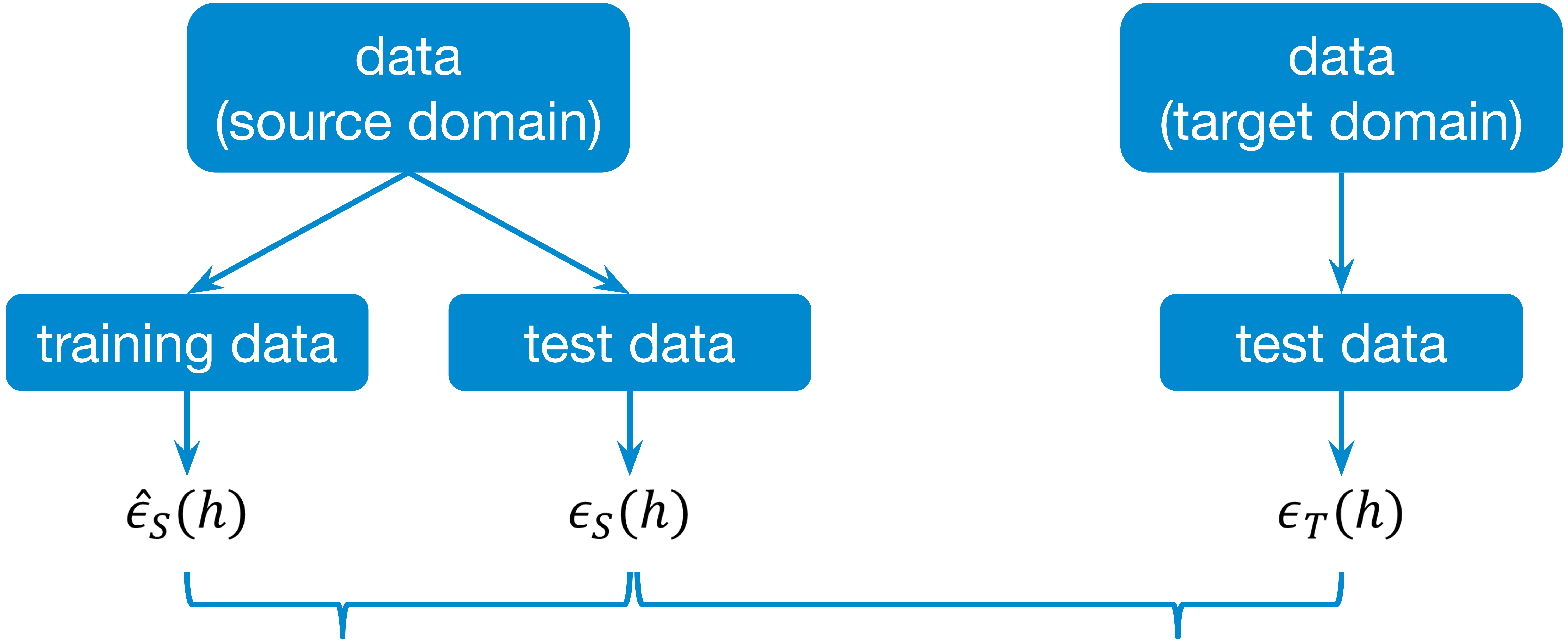
With $1-\delta$ probability, the left inequality holds.

LEARNING FROM SINGLE DOMAIN



$$\epsilon_S(h) \leq \hat{\epsilon}_S(h) + \sqrt{\frac{8}{n}\log\left(\frac{4(2n)^d}{\delta}\right)}$$

LEARNING FROM SINGLE DOMAIN



$$\epsilon_S(h) \leq \hat{\epsilon}_S(h) + \sqrt{\frac{8}{n}\log\left(\frac{4(2n)^d}{\delta}\right)}$$

generalization bound on domain adaptation, i.e., $\epsilon_T(h) \leq ?$

GENERALIZATION BOUND ON DOMAIN ADAPTATION

source domain
D_s, f_s

target domain
D_t, f_t

$\epsilon_S(h)$

$\epsilon_T(h)$

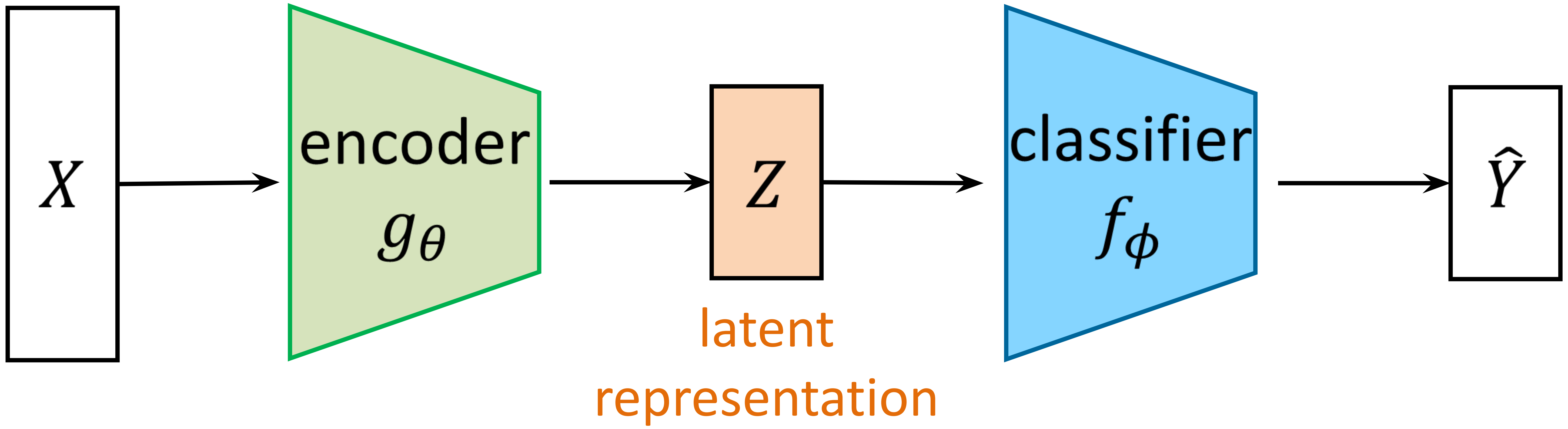
$$\epsilon_T(h) \leq \epsilon_S(h) + d_1(D_S, D_T) + \min\left(\mathbb{E}_{D_S}(|f_S(\boldsymbol{x}) - f_T(\boldsymbol{x})|), \mathbb{E}_{D_T}(|f_S(\boldsymbol{x}) - f_T(\boldsymbol{x})|)\right)$$

the distance between source feature D_S and target feature D_T

the distance between source labeling function f_S and target label function f_T

Ben-David, Shai, et al. "A theory of learning from different domains." Machine learning 79 (2010): 151-175.
https://link.springer.com/content/pdf/10.1007/s10994-009-5152-4.pdf

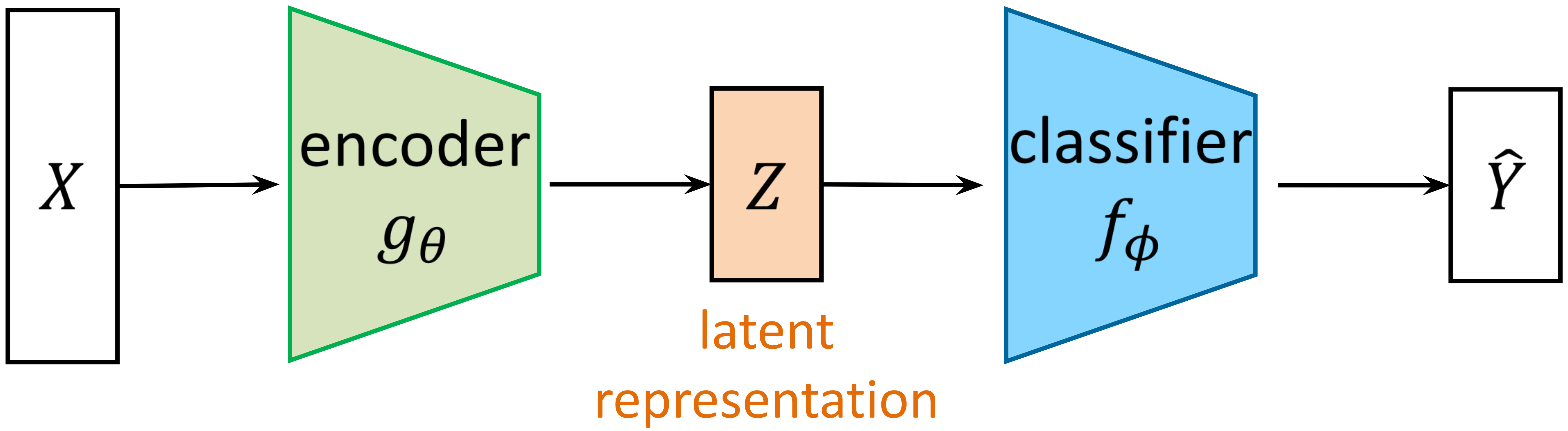
KULLBACK–LEIBLER (KL) DIVERGENCE-GUIDED BOUND



$$\ell_{Test} = \mathbb{E}_{p_t(\boldsymbol{x},y)}[-\log p(\hat{y}|\boldsymbol{x})]$$

Nguyen, A. Tuan, et al. "KL Guided Domain Adaptation." International Conference on Learning Representations. 2021. https://arxiv.org/abs/2106.07780

KULLBACK–LEIBLER (KL) DIVERGENCE-GUIDED BOUND



$$\ell_{Test} = \mathbb{E}_{p_t(\boldsymbol{x},y)}[-\log p(\hat{y}|\boldsymbol{x})] = \mathbb{E}_{p_t(\boldsymbol{x},y)}[-\log \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[p(\hat{y}|\boldsymbol{z})]]$$

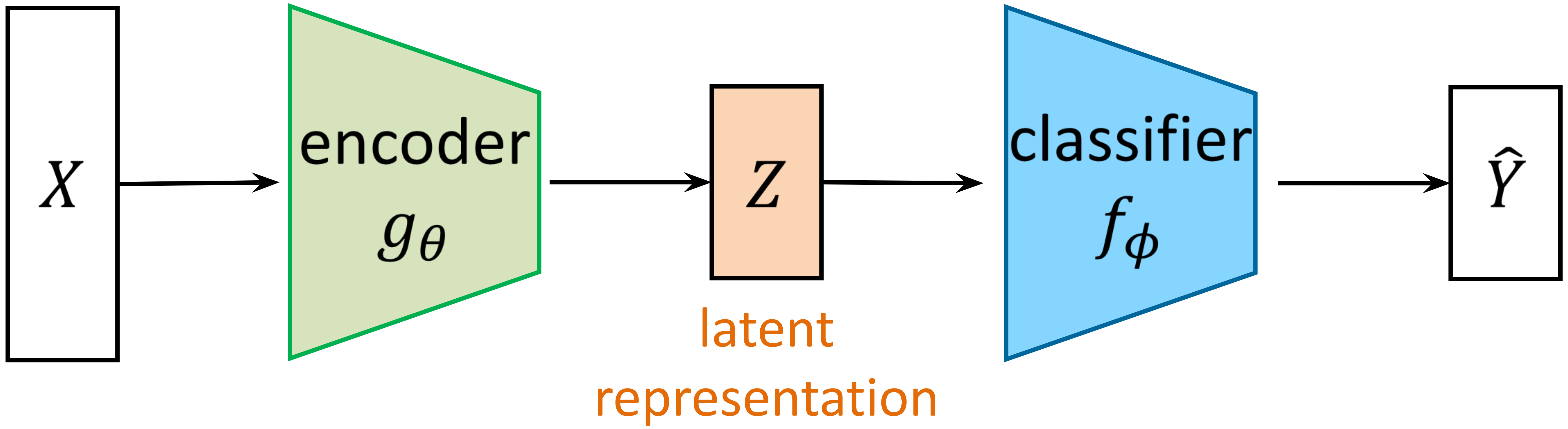
$$\leq \mathbb{E}_{p_t(\boldsymbol{x},y)}\left[\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[-\log p(\hat{y}|\boldsymbol{z})]\right]$$ Jensen inequality

$$= \mathbb{E}_{p_t(\boldsymbol{z},y)}[-\log p(\hat{y}|\boldsymbol{z})]$$

Nguyen, A. Tuan, et al. "KL Guided Domain Adaptation." International Conference on Learning Representations. 2021. https://arxiv.org/abs/2106.07780

VU

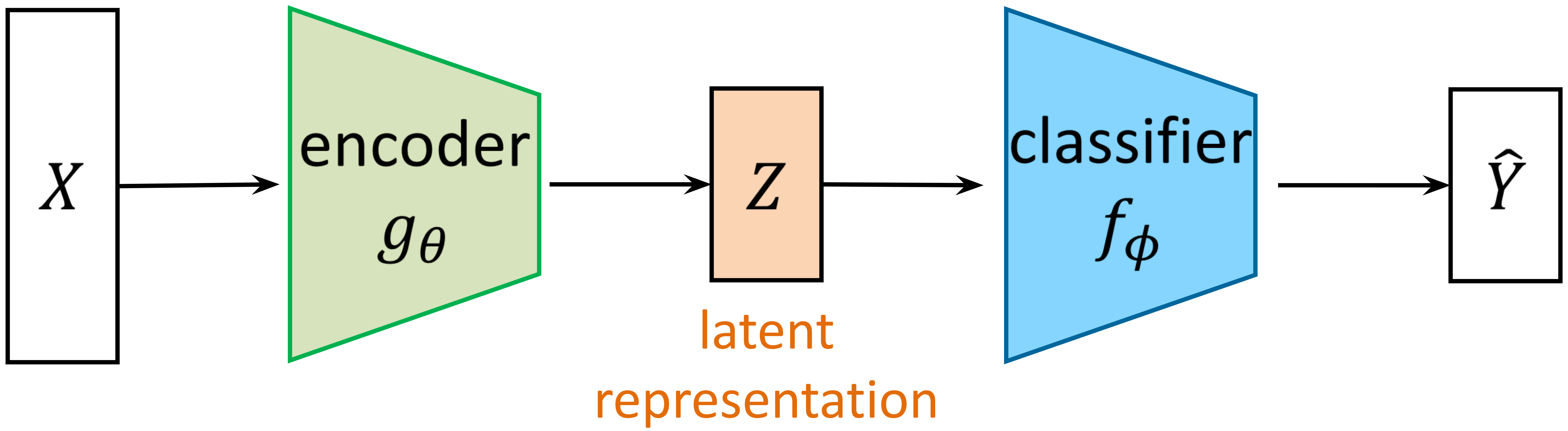
KULLBACK–LEIBLER (KL) DIVERGENCE-GUIDED BOUND



We need to align the joint distribution of $p(\mathbf{z}, y)$, not just $p(\mathbf{z})$!

$$
\begin{aligned}
\ell_{Test} &\leq \mathbb{E}_{p_t(\mathbf{z},y)}[-\log p(\hat{y}|\mathbf{z})] \\
&= \int -\log p(\hat{y}|\mathbf{z}) p_t(\mathbf{z}, y) d\mathbf{z}dy \\
&= \int -\log p(\hat{y}|\mathbf{z}) p_s(\mathbf{z}, y) d\mathbf{z}dy + \int -\log p(\hat{y}|\mathbf{z})[p_t(\mathbf{z}, y) - p_s(\mathbf{z}, y)] d\mathbf{z}dy \\
&= \mathbb{E}_{p_s(\mathbf{z},y)}[-\log p(\hat{y}|\mathbf{z})] + \int -\log p(\hat{y}|\mathbf{z})[p_t(\mathbf{z}, y) - p_s(\mathbf{z}, y)] d\mathbf{z}dy \\
&\leq \ell_{Train} + \frac{M}{2} \int \boxed{|p_t(\mathbf{z}, y) - p_s(\mathbf{z}, y)|} d\mathbf{z}dy
\end{aligned}
$$

$M = \sup -\log p(\hat{y}|\mathbf{z})$



$$\ell_{Test} \leq \ell_{Train} + \frac{M}{2} \int |p_t(\mathbf{z}, y) - p_s(\mathbf{z}, y)| d\mathbf{z} dy$$

$M = \sup - \log p(\hat{y}|\mathbf{z})$

$$\ell_{Test} \leq \ell_{Train} + \frac{M}{\sqrt{2}} \sqrt{\int p_t(\mathbf{z}, y) \log \left(\frac{p_t(\mathbf{z}, y)}{p_s(\mathbf{z}, y)} \right) d\mathbf{z} dy}$$

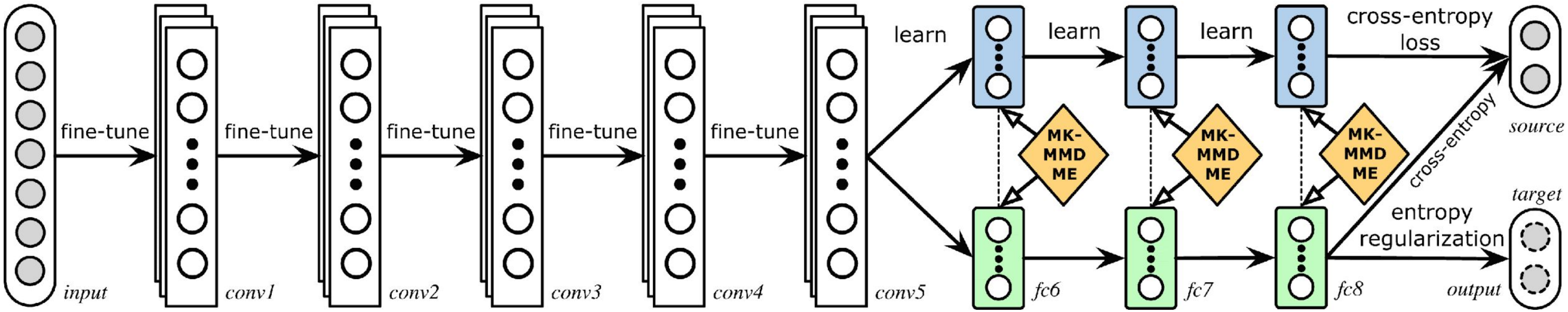
Pinsker's inequality

$$= \ell_{Train} + \frac{M}{\sqrt{2}} \sqrt{D_{KL}(p_t(\mathbf{z}, y); p_s(\mathbf{z}, y))}$$

chain rule of KL divergence

$$= \ell_{Train} + \frac{M}{\sqrt{2}} \sqrt{D_{KL}\big(p_t(\mathbf{z}); p_s(\mathbf{z})\big) + D_{KL}\big(p_t(y|\mathbf{z}); p_s(y|\mathbf{z})\big)}$$

Discrepancy loss in latter layers



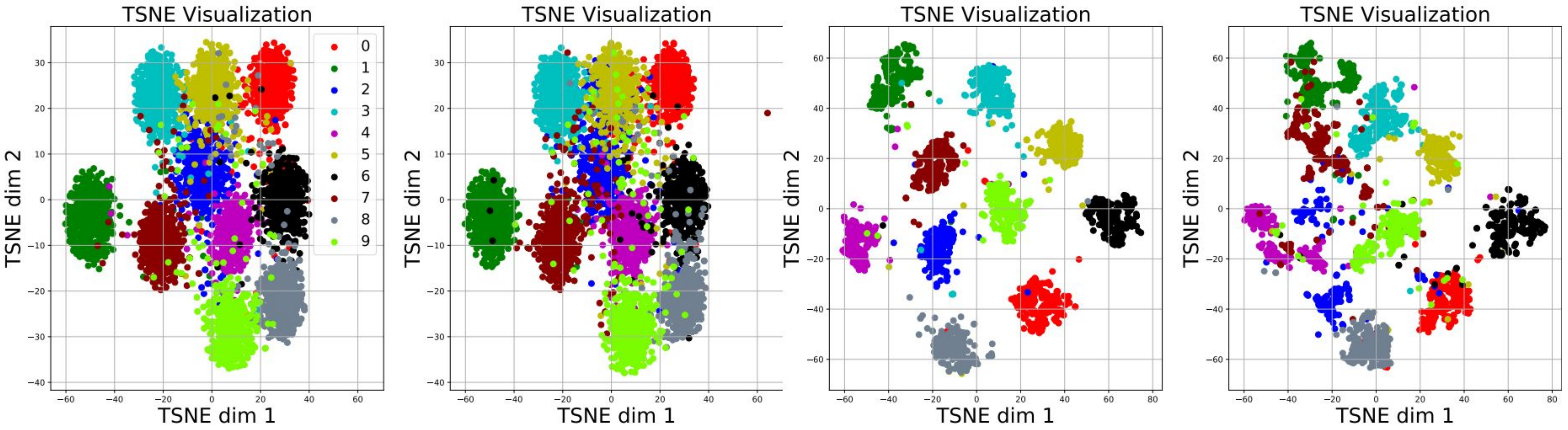
Multi-kernel maximum mean discrepancy(MK-MMD) is defined as

$$d_k^2(p,q) \triangleq \left\| \mathbf{E}_p\left[\phi\left(\mathbf{x}^s\right)\right] - \mathbf{E}_q\left[\phi\left(\mathbf{x}^t\right)\right] \right\|_{\mathcal{H}_k}^2 . \qquad k\left(\mathbf{x}^s, \mathbf{x}^t\right) = \left\langle \phi\left(\mathbf{x}^s\right), \phi\left(\mathbf{x}^t\right) \right\rangle$$

where *phi* is feature mapping function, *k* is kernel function and *H_k* is reproducing kernel Hilbert space

Long, Mingsheng, et al. "Learning transferable features with deep adaptation networks." International conference on machine learning. PMLR, 2015. https://proceedings.mlr.press/v37/long15

KULLBACK–LEIBLER (KL) DIVERGENCE-GUIDED BOUND



(a) KL (b) MMD

Figure 3: Visualization using t-SNE of the representation space of our method KL and the baselines MMD, DANN, ERM. For each method, the left subfigure corresponds to the source domain $\mathcal{M}_0$ and the right one corresponds to the target domain $\mathcal{M}_{45}$. Each color represents a digit class.

PART FOUR: COMPRESSION GENERALIZATION

What is compression? How to design practical algorithms?

INFORMATION BOTTLENECK AND COMPRESSION

Let T be a representation of X

- Which T is useful?

  Disentangled

  Interpretable

INFORMATION BOTTLENECK

Let T be a representation of X

- Which T is useful?

Disentangled

Interpretable

X

"cat" laying on a "laptop"

T

Federici, Marco, et al. "Learning Robust Representations via Multi-View Information Bottleneck." *8th International Conference on Learning Representations*, 2020.

Let T be a representation of X

- Which T is useful?

Disentangled

Interpretable

“cat” laying on a “laptop”

X T

- Tasks

Is there a cat? relevant: “cat”; irrelevant: “laptop”, etc.

How many pixels are there in the image? irrelevant: “cat”, “laptop”

Let T be a representation of X

- Which T is useful?

Disentangled

Interpretable

Related to task Y → Useful for predicting Y

"cat" laying on a "laptop"

X T

- Tasks

Is there a cat? relevant: "cat"; irrelevant: "laptop", etc.

How many pixels are there in the image? irrelevant: "cat", "laptop"

Let T be a representation of X

- Which T is useful?

Disentangled

Interpretable

Related to task Y $\rightarrow$ Useful for predicting Y

- How to define the optimal representation T

Sufficient Statistics $S(X)$

$$\boxed{I(S(X);Y)} = \boxed{I(X;Y)}$$

A representation T of X is sufficient for Y if and only if $I(X;Y) = I(T;Y)$;
T contains all information regarding Y that can be obtained also from X

Let T be a representation of X

- Which T is useful?

  Disentangled

  Interpretable

  Related to task Y $\rightarrow$ Useful for predicting Y

- How to define the optimal representation T

Sufficient Statistics $S(X)$

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

Minimal Sufficient Statistics $T(X)$

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

T contains only relevant information regarding Y, but least information from X.

INFORMATION BOTTLENECK

Let T be a representation of X

- Which T is useful?

  Disentangled

  Interpretable

  Related to task $Y \rightarrow$ Useful for predicting Y

- How to define the optimal representation T

Sufficient Statistics $S(X)$

$$\boxed{I(S(X);Y)} = \boxed{I(X;Y)}$$

Minimal Sufficient Statistics $T(X)$

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

Sufficiency vs Minimality !

Let T be a representation of X

- Which T is useful?

  Disentangled

  Interpretable

  Related to task $Y \rightarrow$ Useful for predicting Y

- How to define the optimal representation T

Sufficient Statistics $S(X)$

$$\boxed{I(S(X);Y)} = \boxed{I(X;Y)}$$

Minimal Sufficient Statistics $T(X)$

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

Information Bottleneck as an Approximation

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

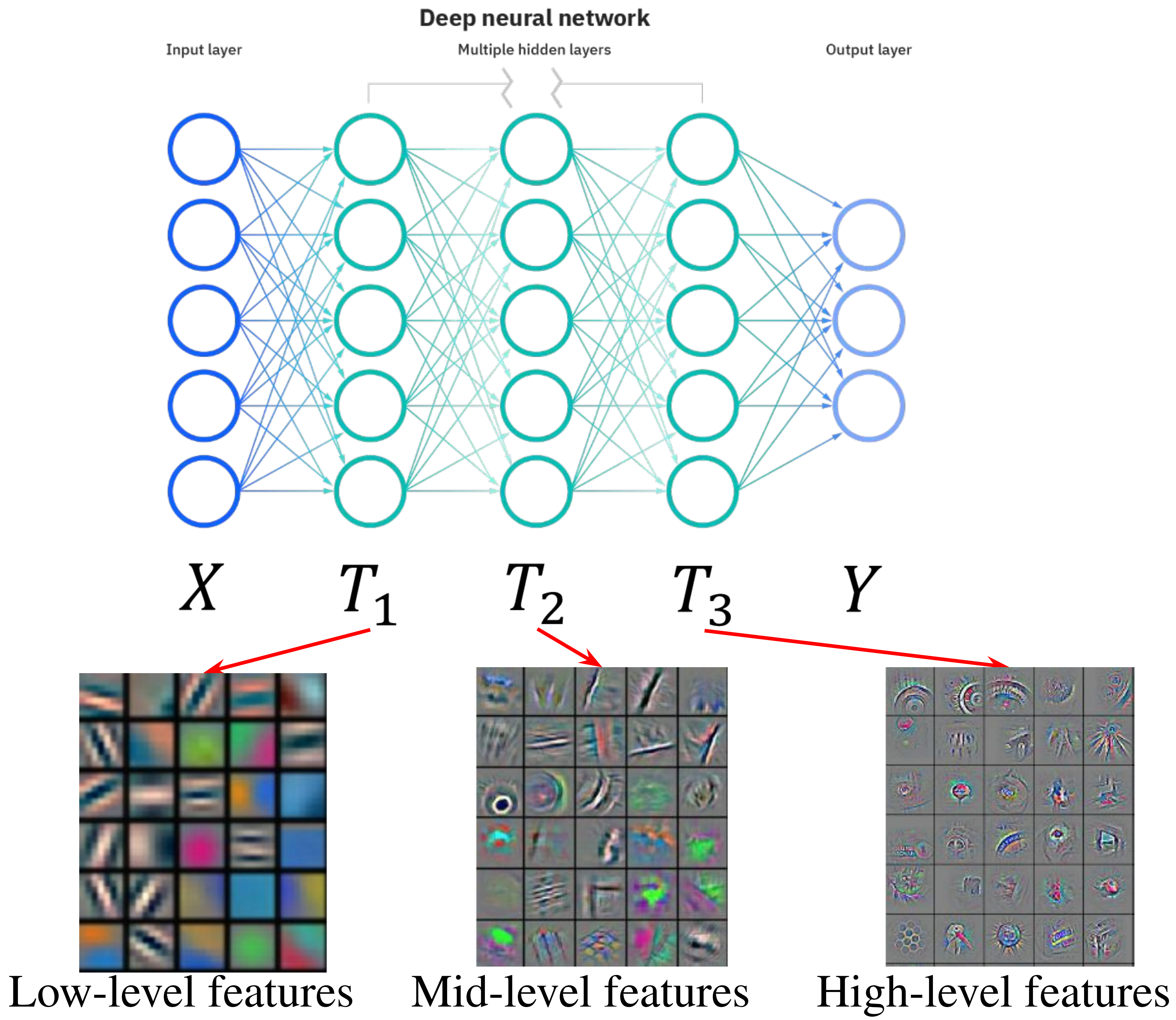
Information Bottleneck

- Given input X and desired output Y, learn a representation T

$$\min_{p(t|x)} \overbrace{I(X;T)}^{\substack{\textbf{\textit{minimality}}\text{ (the complexity}\\ \text{of the representation } T)}} - \beta \underbrace{I(T;Y)}_{\substack{\textbf{\textit{sufficiency}}\text{ (the predictive}\\ \text{performance of } T \text{ on task } Y)}}$$

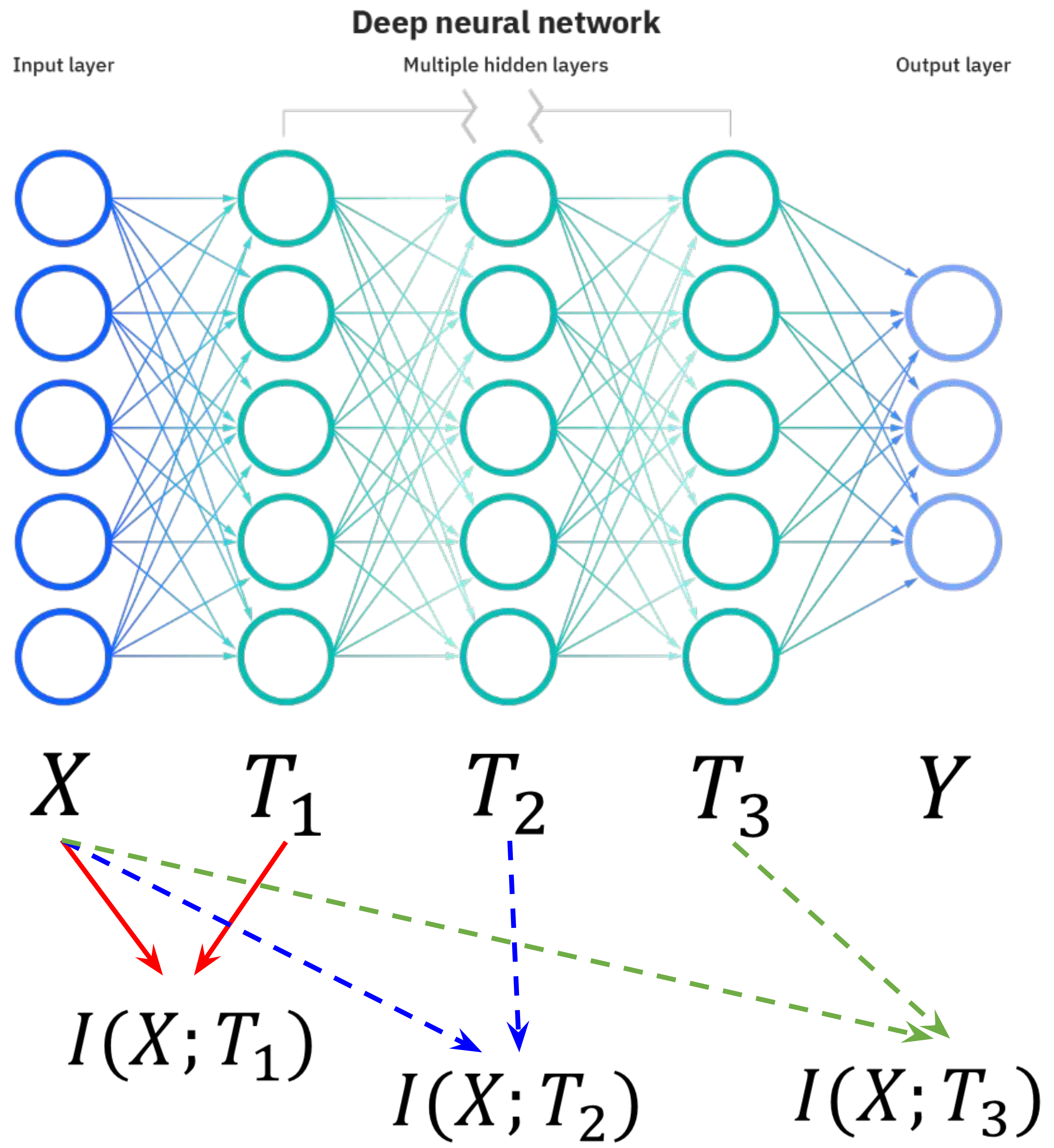
- A natural approximation of minimal sufficient statistic

Tishby, Naftali, Fernando C. Pereira, and William Bialek. "The information bottleneck method." *37th Allerton Conference on Communication and Computation, 2000*.
Gilad-Bachrach, Ran, Amir Navot, and Naftali Tishby. "An information theoretic tradeoff between complexity and accuracy." *Learning Theory and Kernel Machines*. Springer, Berlin, Heidelberg, 2003. 595-609.



DNN as Markov Chain of Random Variables

COMPRESSION IN DEEP LEARNING

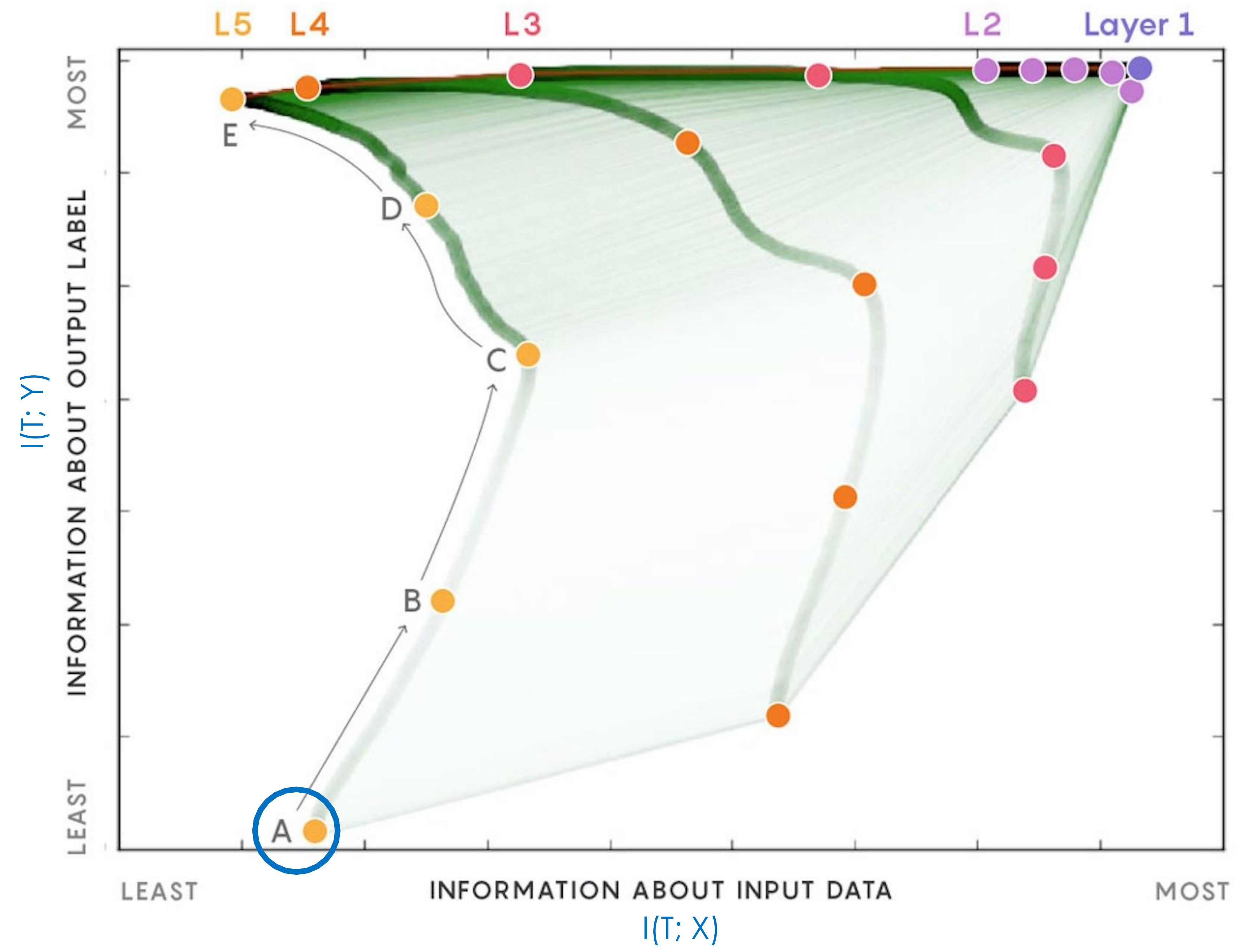


The amount of information that T_i captures about X.

Are there compression in deep neural networks?

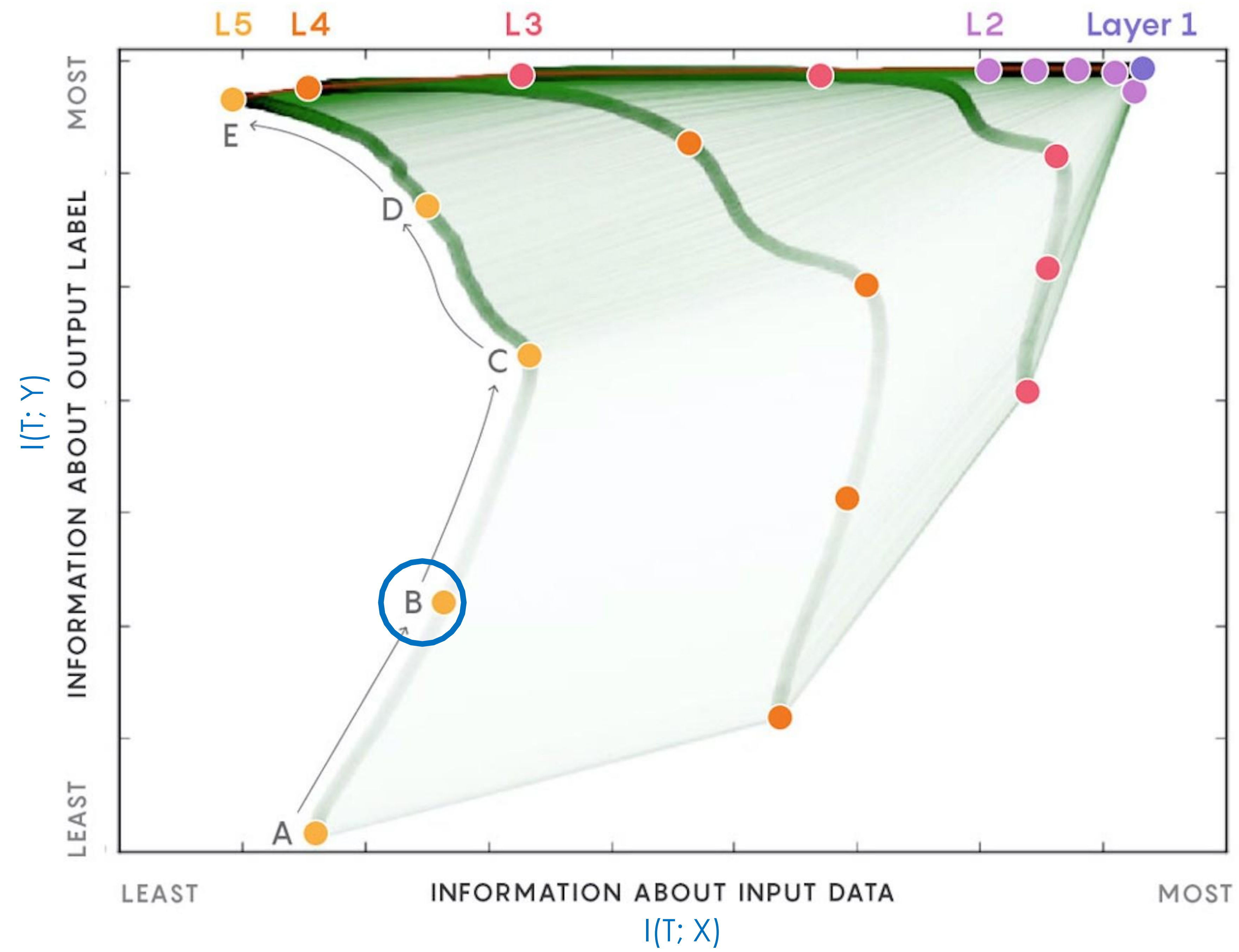
Information Plane: Evolution of $I(T;X)$ $v.s.$ $I(T;Y)$

A Initial State

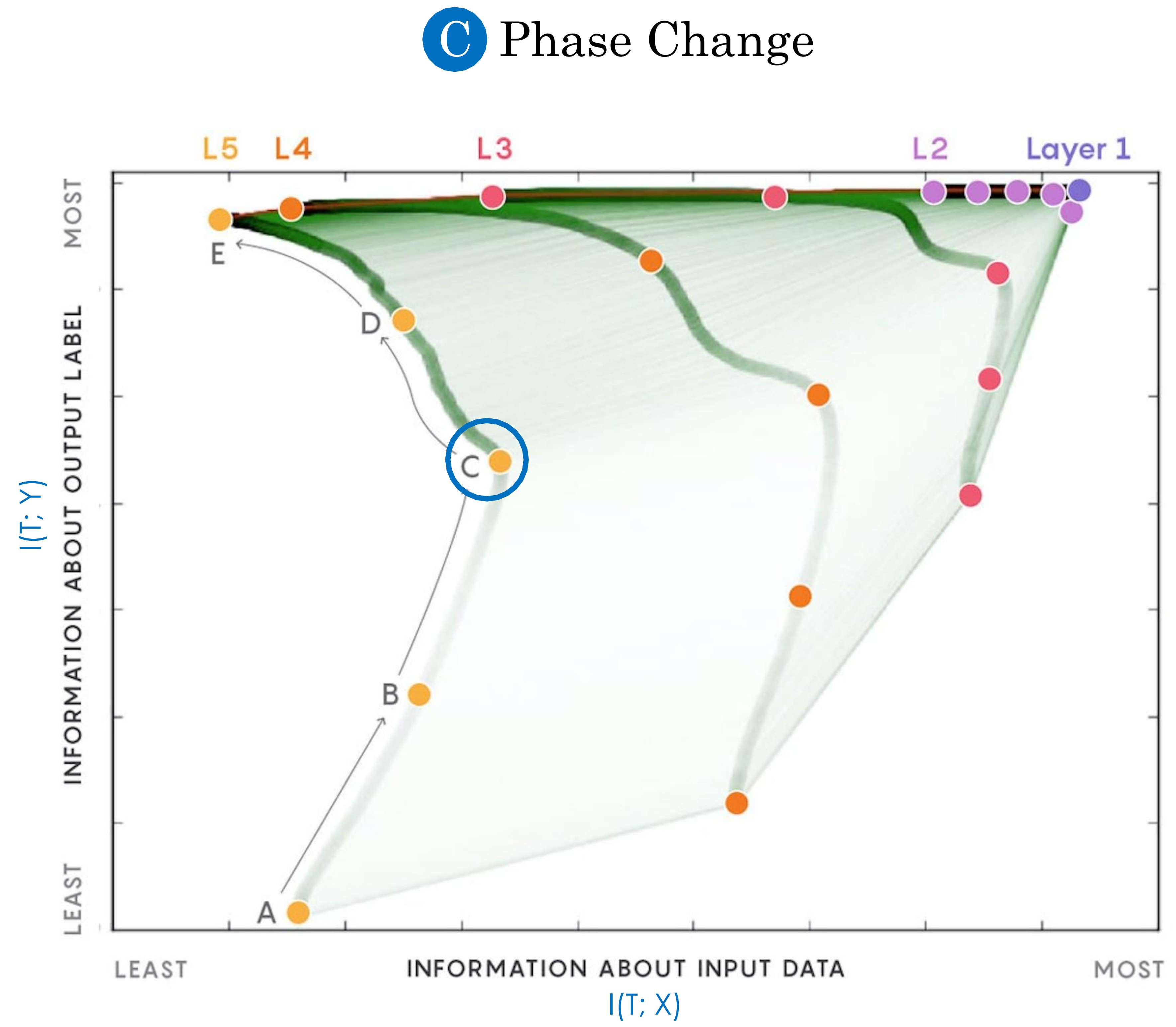


Information Plane: Evolution of $I(T;X)$ v.s. $I(T;Y)$

B Fitting Phase

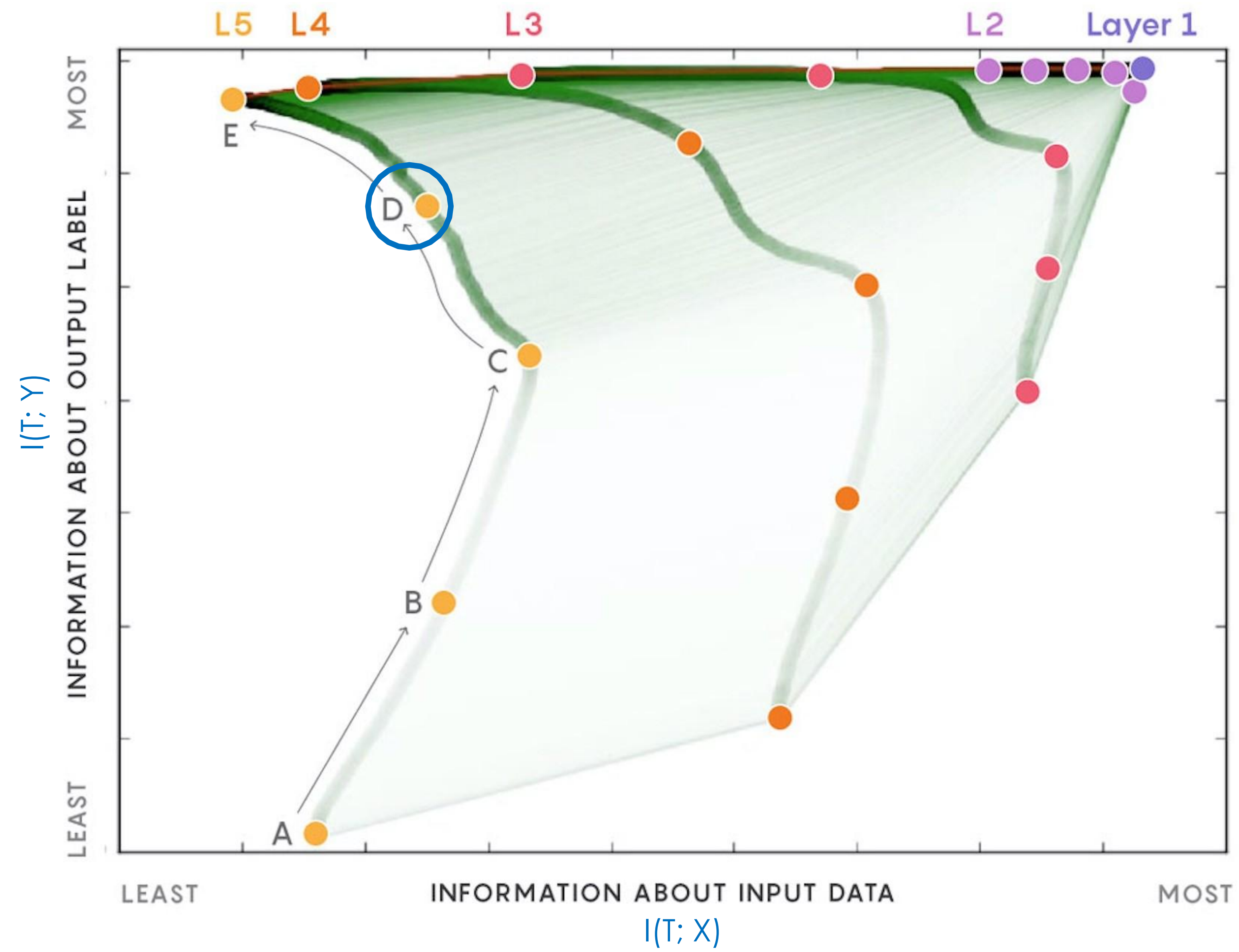


Information Plane: Evolution of $I(T;X)$ v.s. $I(T;Y)$



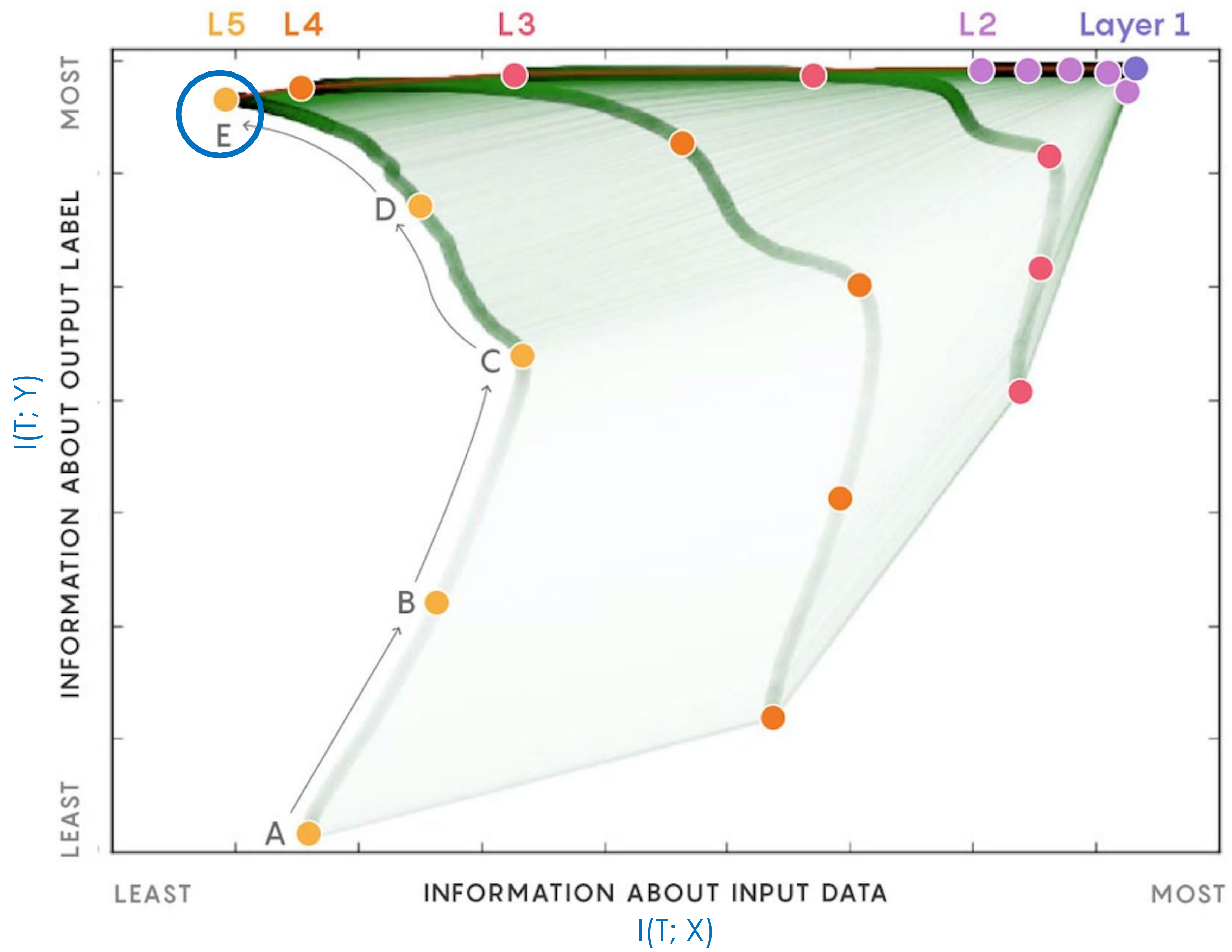
Information Plane: Evolution of $I(T;X)$ v.s. $I(T;Y)$

D Compression Phase



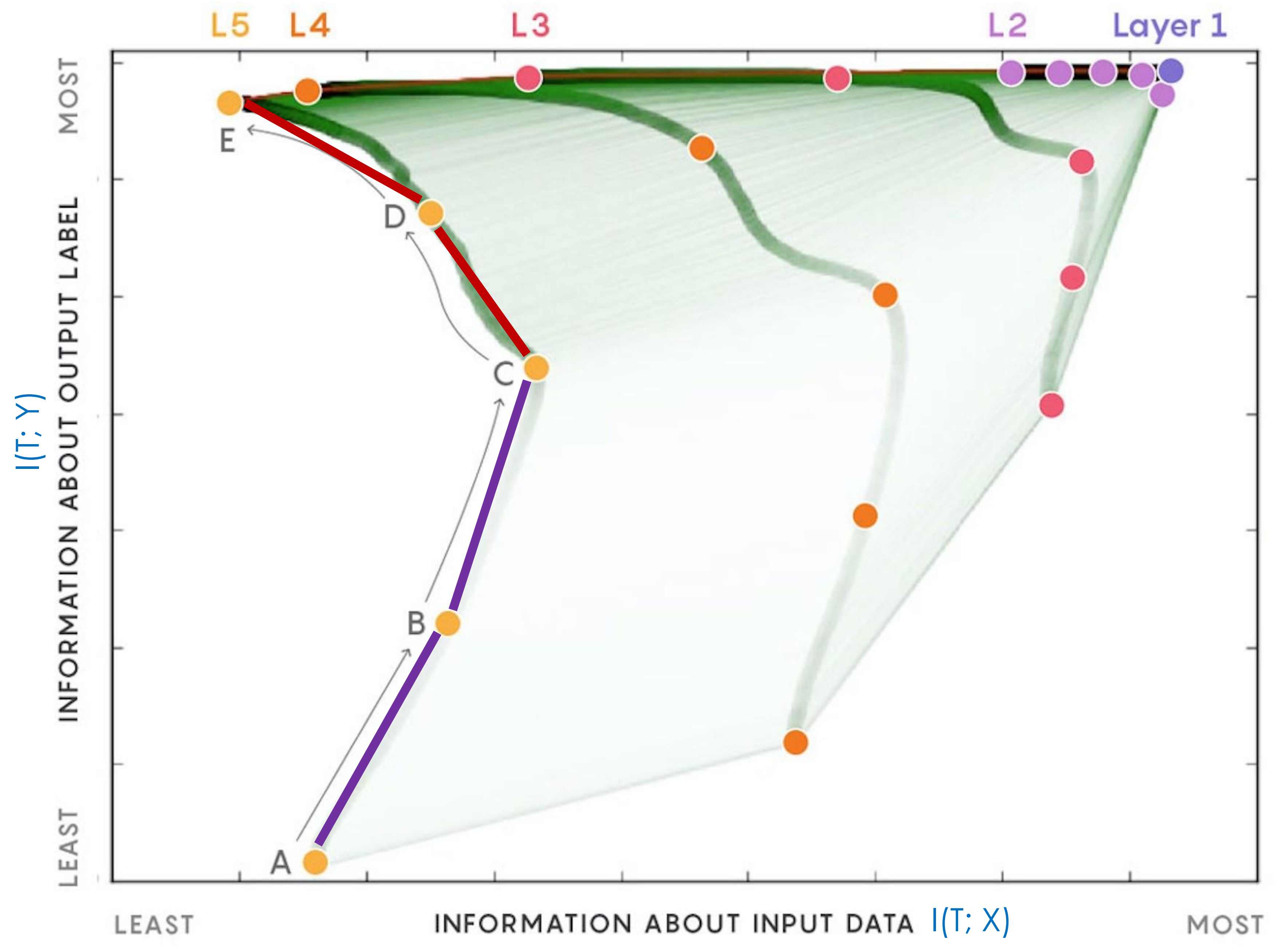
Information Plane: Evolution of $I(T;X)$ v.s. $I(T;Y)$

E Final State



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

- The fitting and compression phase



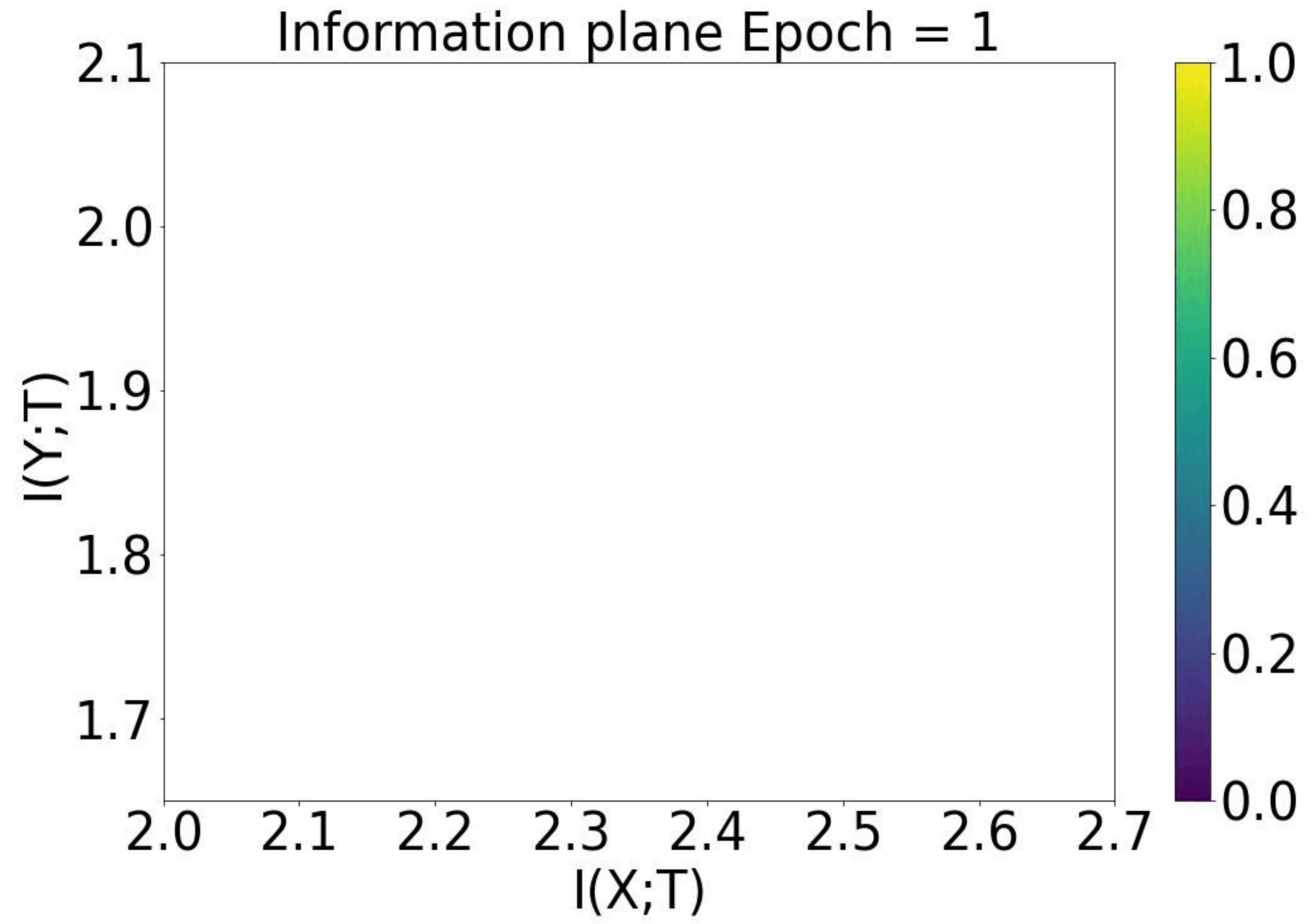
DNN learns in a 2-phase manner:

Phase 1: Information fitting for target

Phase 2: Information compression for sample

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

- The fitting and compression phase

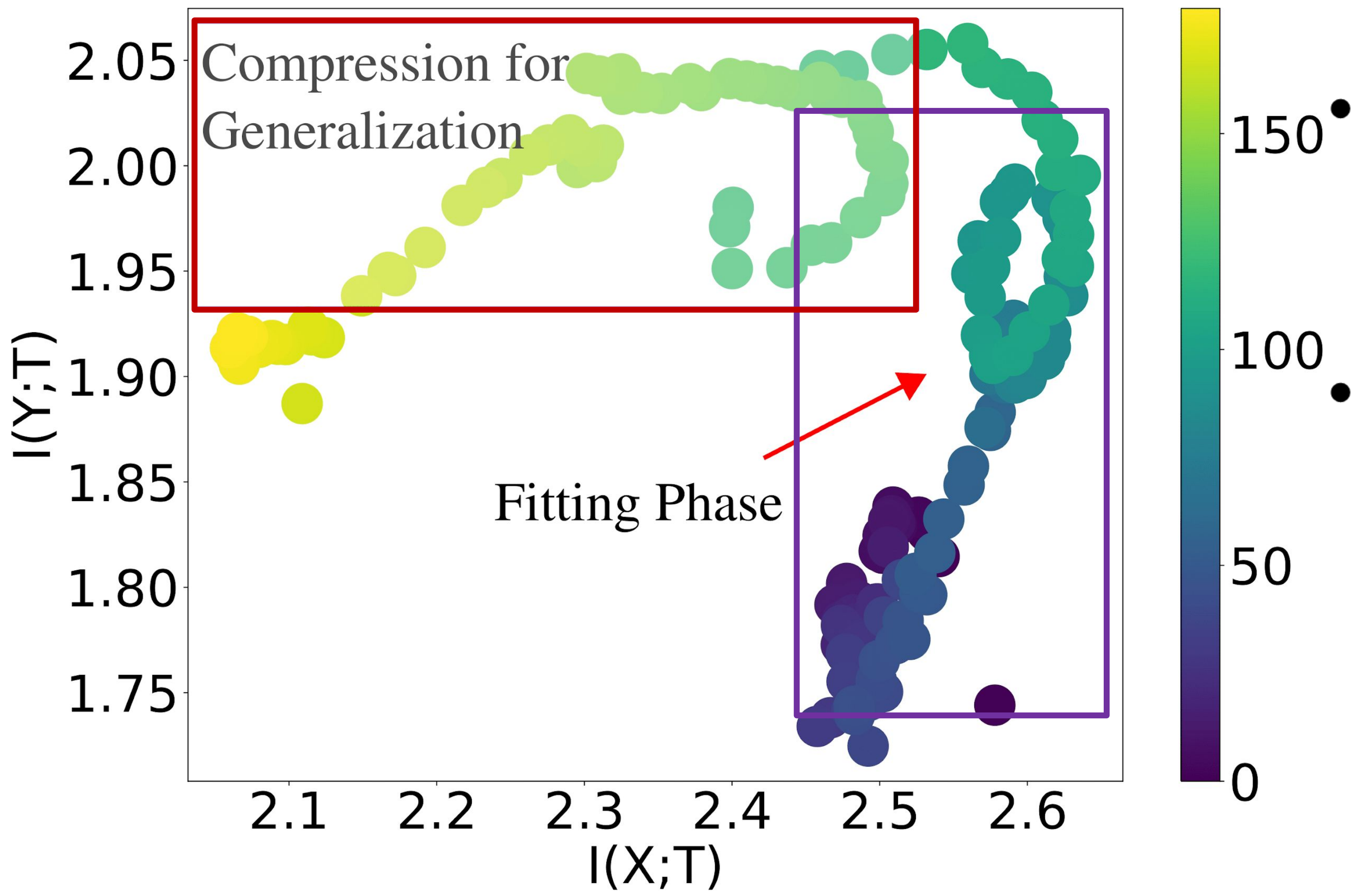


- Information plane.
- $I(X;T)$ wrt. $I(Y;T)$ in each epoch of training

Shwartz-Ziv, Ravid, and Naftali Tishby. "Opening the black box of deep neural networks via information." *arXiv preprint arXiv:1703.00810* (2017).
Yu, Shujian, and Jose C. Principe. "Understanding autoencoders with information theoretic concepts." *Neural Networks* 117 (2019): 104-123.

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

- The fitting and compression phase



The “fitting” and “compression” of latent representations

- Information compression and generalization.
- Final representation reaches to:

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

Information Bottleneck Hypothesis

Shwartz-Ziv, Ravid, and Naftali Tishby. "Opening the black box of deep neural networks via information." *arXiv preprint arXiv:1703.00810* (2017).
Yu, Shujian, and Jose C. Principe. "Understanding autoencoders with information theoretic concepts." *Neural Networks* 117 (2019): 104-123.

Generalization error bound

- Compression implies generalization: for m training samples, with probability at least $1-\delta$,

$$|\mathrm{err}_{\mathrm{train}} - \mathrm{err}_{\mathrm{test}}| < \sqrt{\frac{2^{I(X;T)} + \log(1/\delta)}{2m}}$$

Shwartz-Ziv, Ravid, Amichai Painsky, and Naftali Tishby. "Representation compression and generalization in deep neural networks." https://openreview.net/forum?id=SkeL6sCqK7
Galloway, Angus, et al. "Bounding generalization error with input compression: An empirical study with infinite-width networks." https://openreview.net/forum?id=jbZEUtULft
Kawaguchi, Kenji, et al. "An Analysis of Information Bottlenecks." https://openreview.net/forum?id=h8RIDPvVubq

Deep Information Bottleneck

- $I(X;T)$ as a regularization
- Variational upper bound of $I(X;T)$:

$$
\begin{aligned}
I(T;X) &= \mathbb{E}_{p(x,t)} \log p(t|x) - \mathbb{E}_{p(t)} \log p(t) \\
&\leq \mathbb{E}_{p(x,t)} \log p(t|x) - \mathbb{E}_{p(t)} \log v(t) = D_{\mathrm{KL}}(p(t|x); v(t))
\end{aligned}
$$

Alemi, Alexander A., et al. "Deep variational information bottleneck." *International Conference on Learning Representations*, 2017.

Deep Information Bottleneck

- $I(X;T)$ as a regularization

Model	error
Baseline	1.38%
Dropout	1.34%
Dropout (Pereyra et al., 2017)	1.40%
Confidence Penalty	1.36%
Confidence Penalty (Pereyra et al., 2017)	1.17%
Label Smoothing	1.40%
Label Smoothing (Pereyra et al., 2017)	1.23%
VIB ($\beta = 10^{-3}$)	**1.13%**

Table 1: Test set misclassification rate on permutation-invariant MNIST using $K = 256$. We compare our method (VIB) to an equivalent deterministic model using various forms of regularization. The discrepancy between our results for confidence penalty and label smoothing and the numbers reported in (Pereyra et al., 2017) are due to slightly different hyperparameters.

Alemi, Alexander A., et al. "Deep variational information bottleneck." *International Conference on Learning Representations,* 2017.

Generalization in practical applications

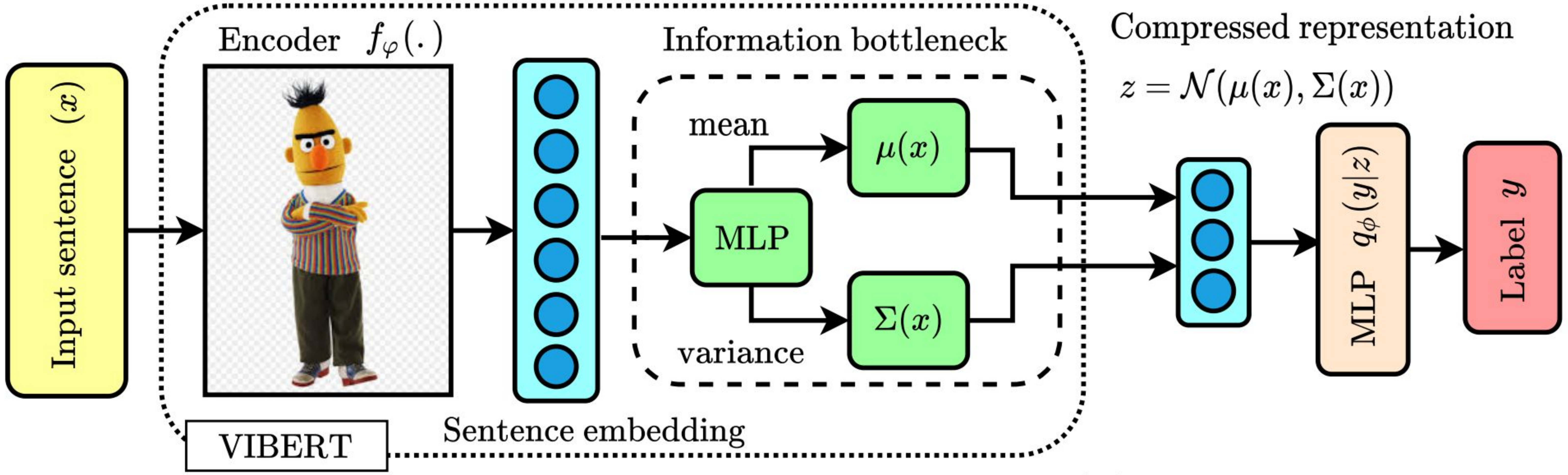


Figure 1: VIBERT compresses the encoder's sentence representation $f_\varphi(x)$ into representation z with mean $\mu(x)$ and eliminates irrelevant and redundant information through the Gaussian noise with variance $\Sigma(x)$.

Mahabadi, Rabeeh Karimi, Yonatan Belinkov, and James Henderson. "Variational Information Bottleneck for Effective Low-Resource Fine-Tuning." *ICLR*, 2021. https://arxiv.org/abs/2106.05469

Generalization in practical applications

Table 1: Average results and standard deviation in parentheses over 3 runs on low-resource data in GLUE. **Δ** shows the absolute difference between the results of the VIBERT model with BERT.

Model	MRPC		STS-B		RTE
	Accuracy	F1	Pearson	Spearman	Accuracy
$\text{BERT}_{\text{Base}}$	87.80 (0.5)	83.20 (0.6)	84.93 (0.1)	83.53 (0.0)	67.93 (1.5)
+Dropout (Srivastava et al., 2014)	87.33 (0.2)	81.90 (0.7)	84.33 (0.9)	82.73(1.0)	65.80 (1.5)
+Mixout (Lee et al., 2019)	87.03 (0.2)	82.63 (0.3)	85.23 (0.4)	83.80(0.4)	67.70 (0.9)
+WD (Lee et al., 2019)	87.57(0.2)	82.83(0.3)	85.0(0.3)	83.6(0.2)	68.63(1.3)
$\text{VIBERT}_{\text{Base}}$	**89.23 (0.1)**	**85.23 (0.2)**	**87.63 (0.3)**	**86.50 (0.4)**	**70.53 (0.5)**
Δ	+1.43	+2.03	+2.7	+2.97	+2.6
$\text{BERT}_{\text{Large}}$	88.47 (0.7)	84.20 (1.3)	86.87 (0.2)	85.70 (0.1)	68.67 (0.8)
+Dropout (Srivastava et al., 2014)	87.77 (0.4)	82.97 (0.2)	86.47 (0.1)	85.33 (0.2)	65.77 (0.6)
+Mixout (Lee et al., 2019)	88.57 (0.7)	84.10 (1.1)	86.70 (0.2)	85.43 (0.3)	70.03 (1.0)
+WD (Lee et al., 2019)	88.97(0.5)	84.87(0.4)	86.9(0.1)	85.67(0.1)	69.27(0.9)
$\text{VIBERT}_{\text{Large}}$	**89.10 (0.4)**	**85.13 (0.6)**	**87.53 (0.8)**	**86.40 (0.9)**	**71.37 (0.8)**
Δ	+0.63	+0.93	+0.66	+0.7	+2.7

Mahabadi, Rabeeh Karimi, Yonatan Belinkov, and James Henderson. "Variational Information Bottleneck for Effective Low-Resource Fine-Tuning." *ICLR*, 2021. https://arxiv.org/abs/2106.05469

THANK YOU FOR YOUR ATTENTION

s.yu3@vu.nl