

Learning deep representations by mutual information estimation and maximization

R Devlon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon,
Karan Grewal, Phil Bachman, Adam Trischler, Yoshua Bengio

Presenter: Seongok Ryu, KAIST Chemistry

Research trends in self-supervised learning

Self-supervised learning

What is self-supervised learning?

“In self-supervised learning, *the system learns to predict part of its input from other parts of its input.*”

In other words, *a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.*”

Self-Supervised Learning: Prediction & Reconstruction Y. LeCun

- ▶ Predict any part of the input from any other part.
- ▶ Predict the **future** from the **past**.
- ▶ Predict the **future** from the **recent past**.
- ▶ Predict the **past** from the **present**.
- ▶ Predict the **top** from the **bottom**.
- ▶ Predict the **occluded** from the **visible**
- ▶ **Pretend there is a part of the input you don't know and predict that.**

Time →

← Past Present Future →


– Yann LeCun, 2019.

Self-supervised learning

Why is it becoming a new research trend?

How Much Information is the Machine Given during Learning? Y. LeCun

- ▶ **“Pure” Reinforcement Learning (cherry)**
 - ▶ The machine predicts a scalar reward given once in a while.
 - ▶ **A few bits for some samples**
- ▶ **Supervised Learning (icing)**
 - ▶ The machine predicts a category or a few numbers for each input
 - ▶ Predicting human-supplied data
 - ▶ **10→10,000 bits per sample**
- ▶ **Self-Supervised Learning (cake génoise)**
 - ▶ The machine predicts any part of its input for any observed part.
 - ▶ Predicts future frames in videos
 - ▶ **Millions of bits per sample**



Self-supervised learning

Unsupervised vs Self-supervised

- Unsupervised learning have been commonly performed by density estimation by using such as autoencoders and generative adversarial networks.
- It aims to perform density estimation, generation of new instances, and representation learning.
- By quoting prof. Yann Lecun:

“Self-supervised learning uses way more supervisory signals than supervised learning, and enormously more than reinforcement learning. That's why calling it "unsupervised" is totally misleading. That's also why more knowledge about the structure of the world can be learned through self-supervised learning than from the other two paradigms: the data is unlimited, and amount of feedback provided by each example is huge.”

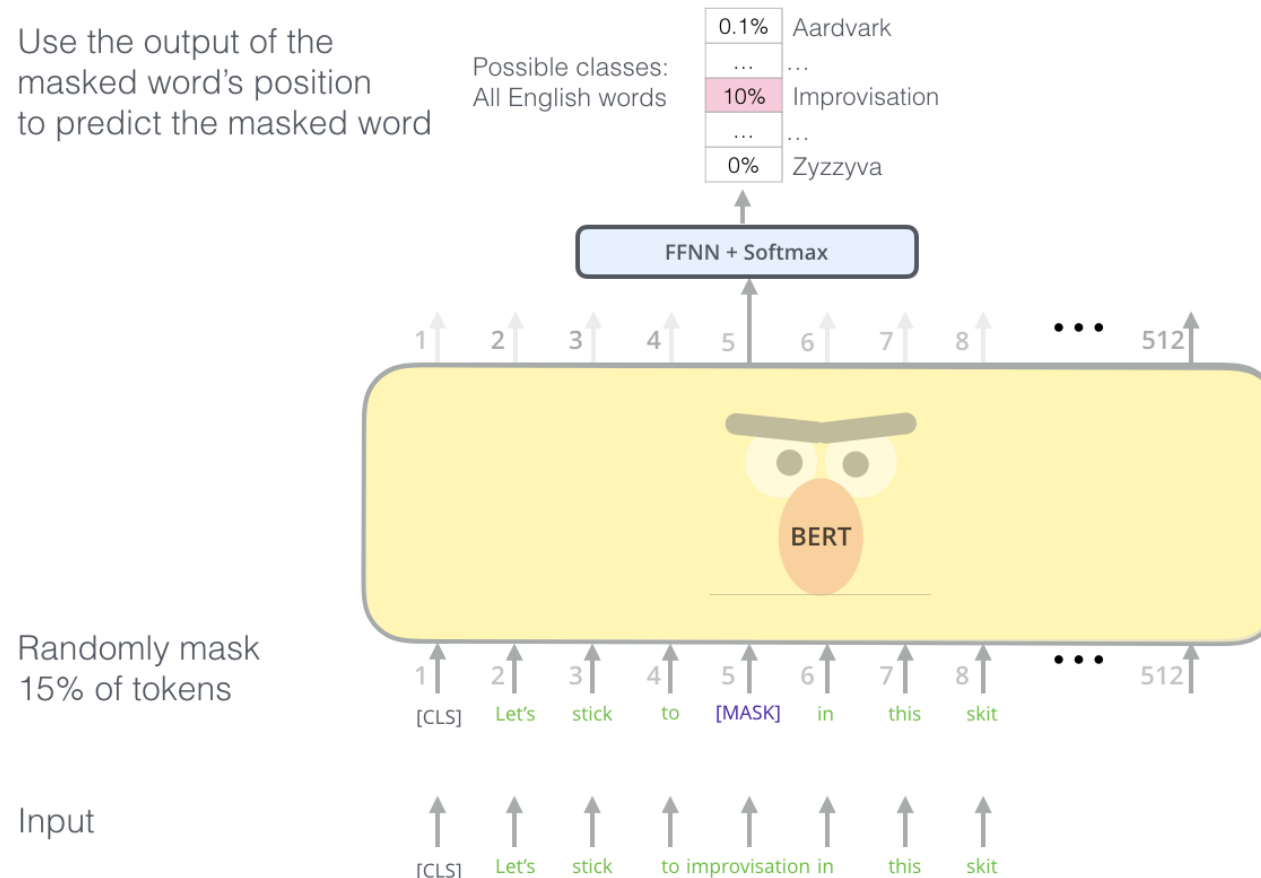
- Yann LeCun, 2019.

Self-supervised learning

Current state-of-the-arts self-supervised learning algorithms.

BERT : predicting the output of the masked words.

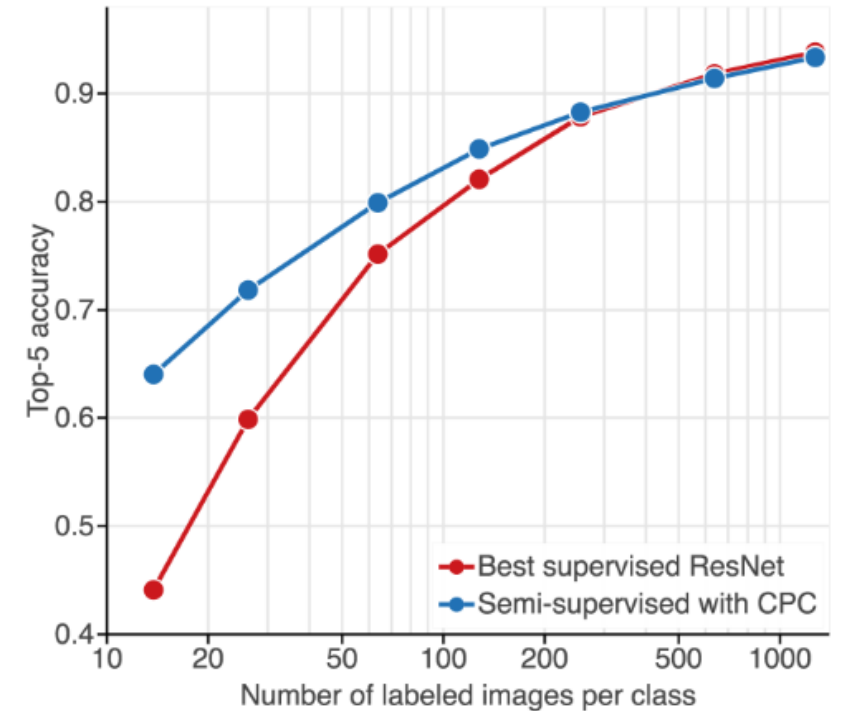
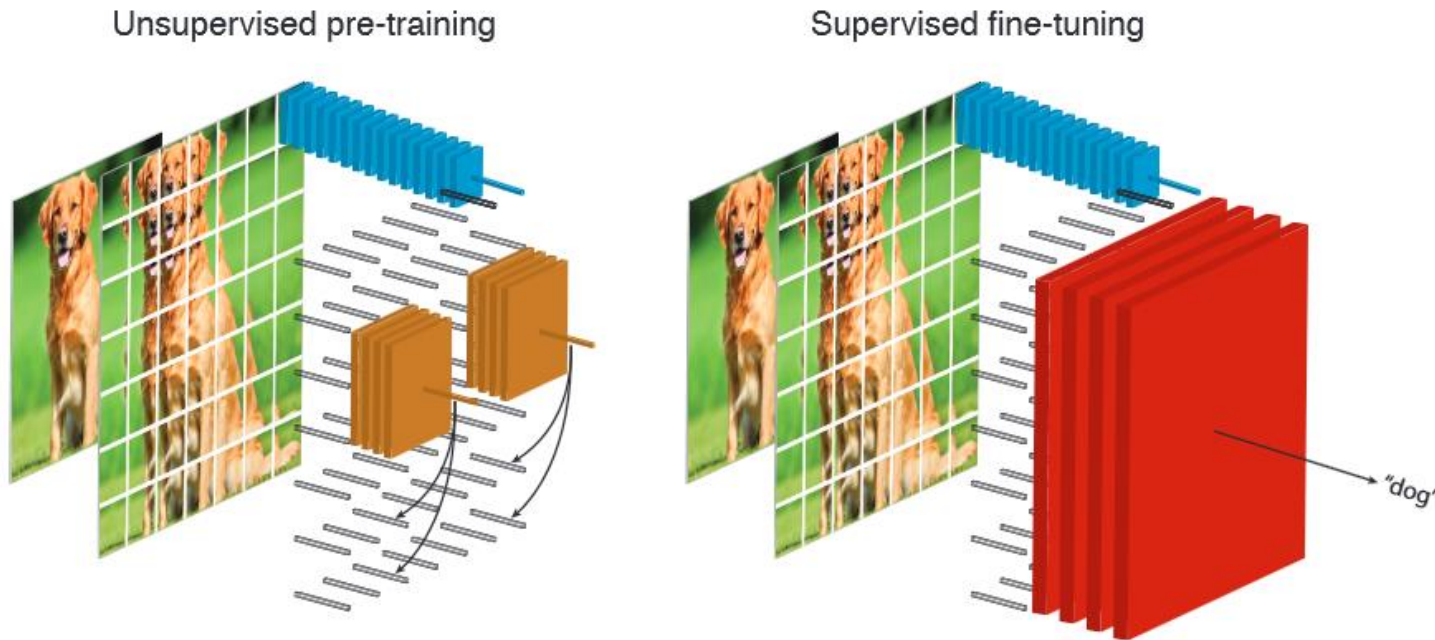
→ Masked Language Modeling (MLM)



Self-supervised learning

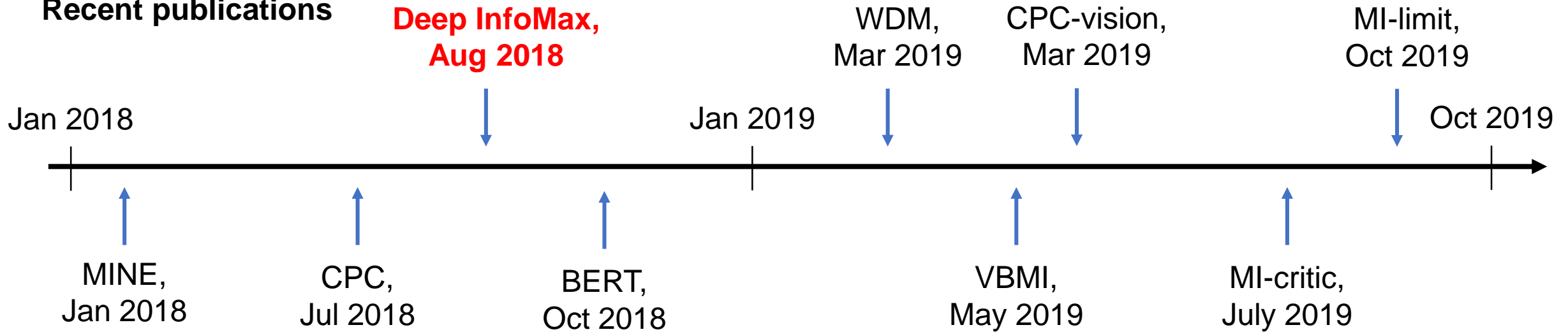
Current state-of-the-arts self-supervised learning algorithms.

Contrastive predictive coding: minimize the objective (infoNCE loss) to predict the other image patches



Self-supervised learning

Recent publications



- MINE: Mutual Information Neural Estimation
- CPC: Representation Learning with Contrastive Predictive Coding
- Deep InfoMax: Learning deep representations by mutual information estimation and maximization
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
- WDM: Wasserstein Dependency Measure for Representation Learning
- VBMI: On Variational Bounds of Mutual Information
- CPC-vision: Data-Efficient Image Recognition with Contrastive Predictive Coding
- MI-critic: On Mutual Information Maximization for Representation Learning
- MI-limit: Understanding the Limitations of Variational Mutual Information Estimator

Deep InfoMax

<https://www.microsoft.com/en-us/research/blog/deep-infomax-learning-good-representations-through-mutual-information-maximization/>

<https://github.com/rdevon/DIM>

Motivations

Representation learning by infoMax principle

- InfoMax principle: maximizing the mutual information between the inputs X and its representation Z to yield good representation learning

$$\max_{\theta \in \Theta} I_{\theta}(X; Z)$$



X : input images



Z : representations

Motivations

Challenges in using InfoMax principle and proposed solutions

1. Exact computation of mutual information, $I(X; Z) = \text{KL}(p(X, Z) || p(X)p(Z))$, is mostly intractable.

→ Maximizing **a tractable (variational) lower bound** on the quantity

$$I(X; Z) \geq \hat{I}_{\Theta}(X, Z)$$

→ (MINE) For example, using the Donsker-Varadhan (DV) representation, the dual representation of KL-divergence, enables us to estimate the mutual information **by sampling random variables**:

$$\hat{I}_{\text{DV}}(X, Z) = \sup_{\theta \in \Theta} \mathbb{E}_{p(X, Z)}[T_{\theta}] - \log \mathbb{E}_{p(X)p(Z)}[e^{T_{\theta}}]$$

→ (CPC) minimizing the infoNCE loss is equivalent to maximizing **the lower bound of the mutual information**

$$\hat{I}_{\text{infoNCE}}(X, Z) = \mathbb{E}_{p(X, Z)}[f(X, Z)] - \mathbb{E}_{p(X)} \left[\log \mathbb{E}_{p(Z)} [\exp f(X, Z)] \right]$$

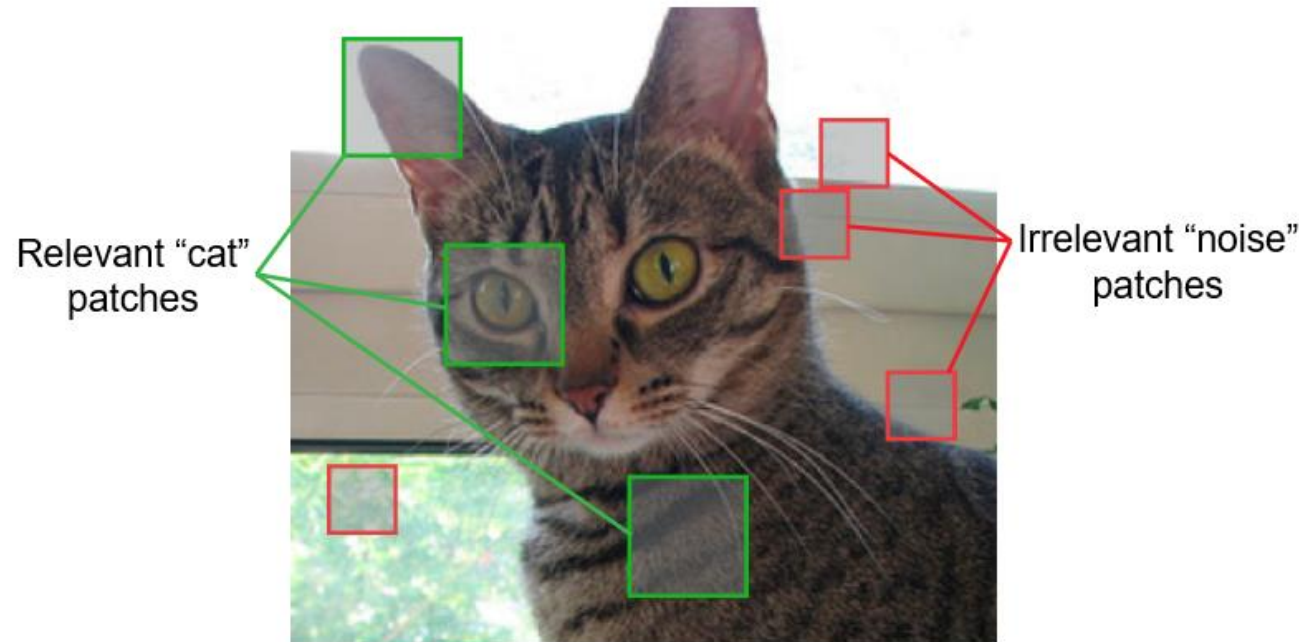
Motivations

Challenges in using InfoMax principle and proposed solutions

2. We are only interested in relevant features:

“maximizing mutual information between the whole image input and the output representation, which we refer to as global DIM in the paper, *will be biased toward learning features that are unrelated*, as their sum has more unique information than redundant locations.”

<https://www.microsoft.com/en-us/research/blog/deep-infomax-learning-good-representations-through-mutual-information-maximization/>



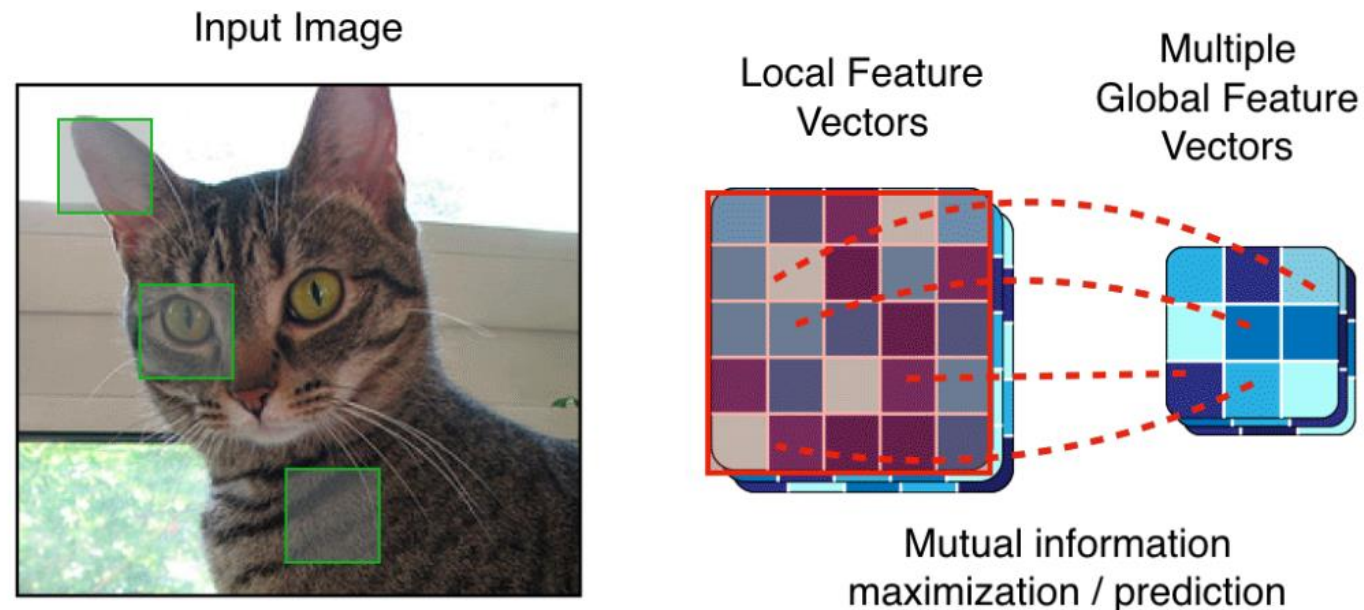
Motivations

Challenges in using InfoMax principle and proposed solutions

2. Multi-view coding: maximizing the MI between local and global features, $g_1(X^{(1)})$ and $g_2(X^{(2)})$, where $X^{(1)}$ and $X^{(2)}$ is different, possibly overlapping view of X , instead of raw sensory inputs and global features.

$$\max_{g_1 \in \mathcal{G}_1, g_2 \in \mathcal{G}_2} \hat{I}(g_1(X^{(1)}); g_2(X^{(2)}))$$

$\because I(X; Y) = I(X'; Y')$, if $X' = f_1(X)$ and $Y' = f_2(Y)$ are homeomorphisms (smooth invertible maps).



Motivations

What Deep InfoMax studied?

- Simultaneously estimates and maximizes the mutual information between local and global representations.
- Suggesting a number of lower bounds of the mutual information,
→ “It is unnecessary to use the exact KL-based formulation of MI.”

Training criteria

- **Mutual information maximization:** Find the set of parameters, ψ , such that the mutual information $I(X; E_{\psi}(X))$ is maximized.
- **Statistical constraints:** the marginal $\mathbb{U}_{\psi, \mathbb{P}}$ should match a prior distribution \mathbb{V} . Roughly speaking, this can be used to encourage the output of the encoder to have desired characteristics (e.g., independence).

Idea

The most basic idea

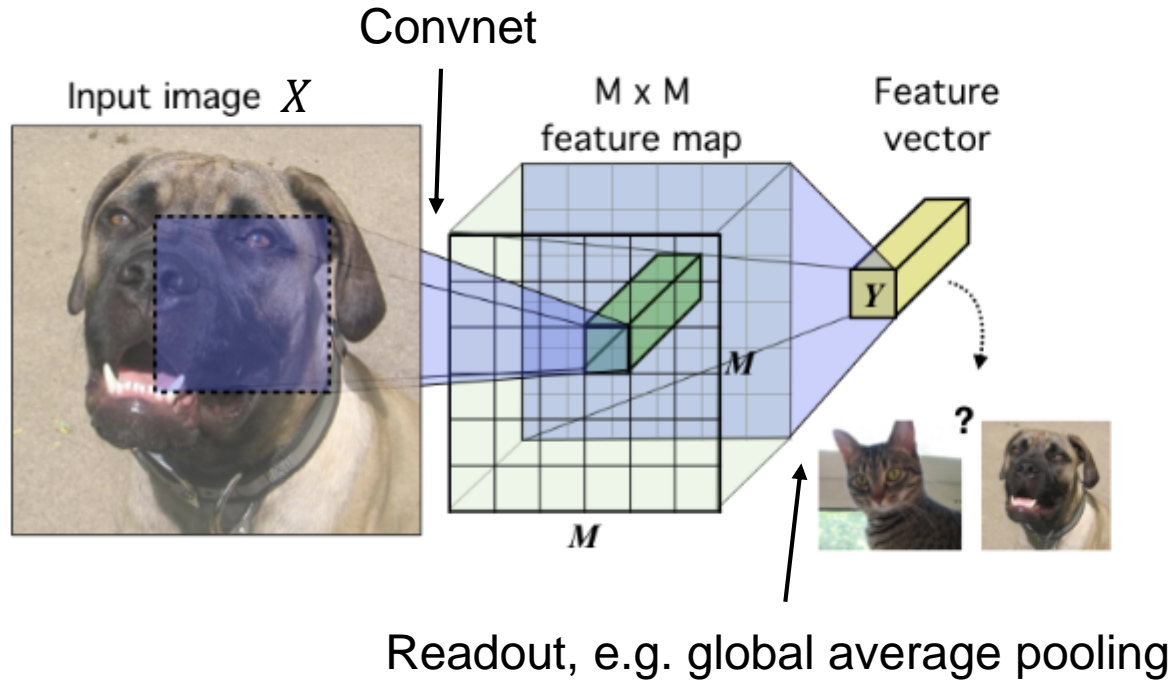


Figure 1: **The base encoder model in the context of image data.** An image (in this case) is encoded using a convnet until reaching a feature map of $M \times M$ feature vectors corresponding to $M \times M$ input patches. These vectors are summarized into a single feature vector, Y . Our goal is to train this network such that useful information about the input is easily extracted from the high-level features.

- $X := \{x^{(i)} \in \mathcal{X}\}_{i=1}^N$ with empirical probability distribution \mathbb{P} .
- $E_\psi: \mathcal{X} \rightarrow \mathcal{Y}$ with parameters ψ (e.g., a neural network) is an element of a family of encoders $\{E_\psi\}_{\psi \in \Psi}$ over Ψ .
- $\mathbb{U}_{\psi, \mathbb{P}}$ is the distribution over encodings $y \in \mathcal{Y}$ produced by sampling observations $x \sim \mathcal{X}$ and then sampling $y \sim E_\psi(x)$

Global DIM framework

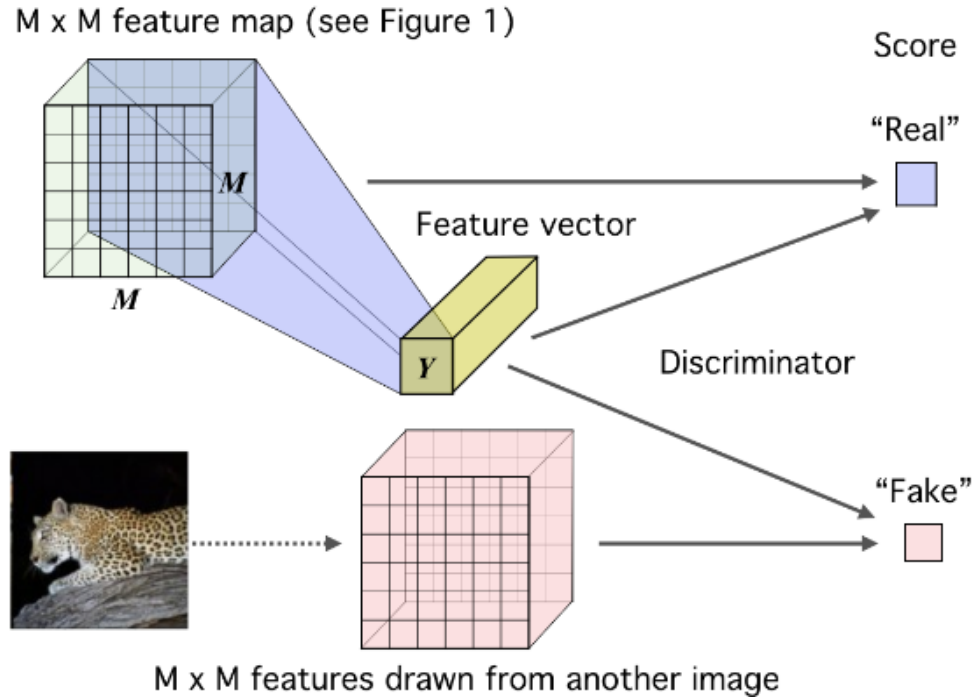


Figure 2: **Deep InfoMax (DIM) with a global $MI(X; Y)$ objective.** Here, we pass both the high-level feature vector, Y , and the lower-level $M \times M$ feature map (see Figure 1) through a discriminator to get the score. Fake samples are drawn by combining the same feature vector with a $M \times M$ feature map from another image.

- Discriminate whether the (true/query) feature vector comes from the positive sample or other negative samples.
- This framework can be implemented by using binary cross-entropy loss.

Idea

Local DIM framework

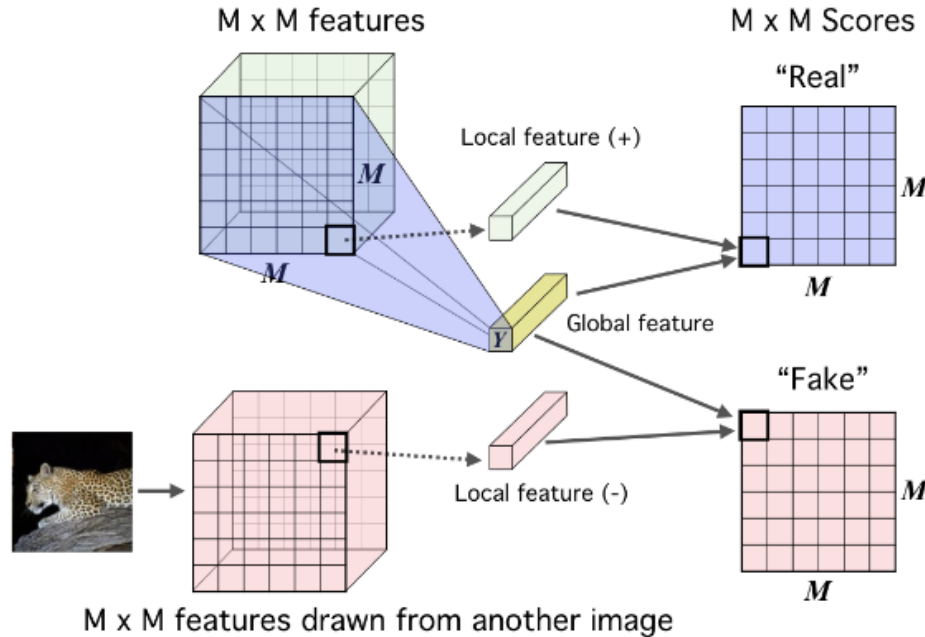


Figure 3: **Maximizing mutual information between local features and global features.**

First we encode the image to a feature map that reflects some structural aspect of the data, e.g. spatial locality, and we further summarize this feature map into a global feature vector (see Figure 1). We then concatenate this feature vector with the lower-level feature map *at every location*. A score is produced for each local-global pair through an additional function (see the Appendix A.2 for details).

- Encoding the input to a feature map, $C_\psi(x) := \{C_\psi^{(i)}\}_{i=1}^{M \times M}$ that reflects useful structure in the data
- Maximizing the average estimated MI:

$$(\hat{\omega}, \hat{\psi})_L = \operatorname{argmax}_{\omega, \psi} \frac{1}{M^2} \sum_{i=1}^{M^2} \hat{I}_{\omega, \psi} \left(C_\psi^{(i)}(X); E_\psi(X) \right)$$

Local feature
Global feature

Variational MI objectives for Local DIM framework

Objective 1) Following the mutual information neural estimation (MINE)

- For two random variables sampled from the joint distribution $(x, y) \sim \mathbb{J}$ and the product of its marginal are \mathbb{M} ,
- MINE uses a lower-bound to the MI based on the Donsker-Varadhan representation (DV) of the KL-divergence,

$$I(X; Y) = \mathcal{D}_{KL}(\mathbb{J} \parallel \mathbb{M}) \geq \hat{I}_{\omega}^{(DV)}(X; Y) := \mathbb{E}_{\mathbb{J}}[T_{\omega}(x, y)] - \log \mathbb{E}_{\mathbb{M}}[e^{T_{\omega}(x, y)}],$$

where $T_{\omega}: \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is a discriminator function (a statistics network) is modeled by a neural network ω .

- At a high level, we optimize E_{ψ} by simultaneously estimating and maximizing $I(X; E_{\psi}(X))$:

$$(\hat{\omega}, \hat{\psi})_G = \operatorname{argmax}_{\omega, \psi} \hat{I}_{\omega}^{(DV)}(X; E_{\psi}(X))$$

Variational MI objectives for Local DIM framework

Objective 2) Using a Jensen-Shannon MI estimator

- **Since we are primarily interested in maximizing MI, and not concerned with its precise value,** one could define a Jensen-Shannon MI estimator:

$$\hat{I}_{\omega, \psi}^{(\text{JSD})}(X; E_{\psi}(X)) := \mathbb{E}_{\mathbb{P}} \left[-\text{sp} \left(-T_{\psi, \omega} \left(x, E_{\psi}(x) \right) \right) \right] - \mathbb{E}_{\mathbb{P} \times \tilde{\mathbb{P}}} \left[\text{sp} \left(T_{\psi, \omega} \left(x', E_{\psi}(x) \right) \right) \right],$$

where x is an (positive) input sample, x' is an (negative) input sampled from $\tilde{\mathbb{P}} = \mathbb{P}$, and $\text{sp}(z) = \log(1 + e^z)$.

- $E_{\psi} = f_{\psi} \circ C_{\psi}$ and $T_{\psi, \omega} = D_{\omega} \circ g \circ (C_{\psi}, E_{\psi})$
- C_{ψ} maps a $M \times M$ input to $M \times M$ a (local) feature map,

f_{ψ} summarized this feature map to a global feature,

g is a function that combines the encoder output with the lower layer, and

D_{ω} is a discriminator function parameterized by ω .

Variational MI objectives for Local DIM framework

Objective 3) Noise-contrastive estimation (NCE)

- InfoNCE loss can also be used with DIM by maximizing:

$$\hat{I}_{\omega, \psi}^{(\text{infoNCE})} (X; E_{\psi}(X)) := \mathbb{E}_{\mathbb{P}} \left[T_{\psi, \omega} (x, E_{\psi}(x)) - \mathbb{E}_{\tilde{\mathbb{P}}} \left[\log \sum_{x'} e^{T_{\psi, \omega}(x', E_{\psi}(x))} \right] \right].$$

- For DIM, a key difference between the DV, JSD, and infoNCE formulations is whether an expectation over $\mathbb{P}/\tilde{\mathbb{P}}$ appears inside or outside of a log.

Comparison of variational MI objectives

- We found that DIM with the JSD loss is insensitive to the number of negative samples, and in fact outperforms infoNCE as the number of negative samples becomes smaller.

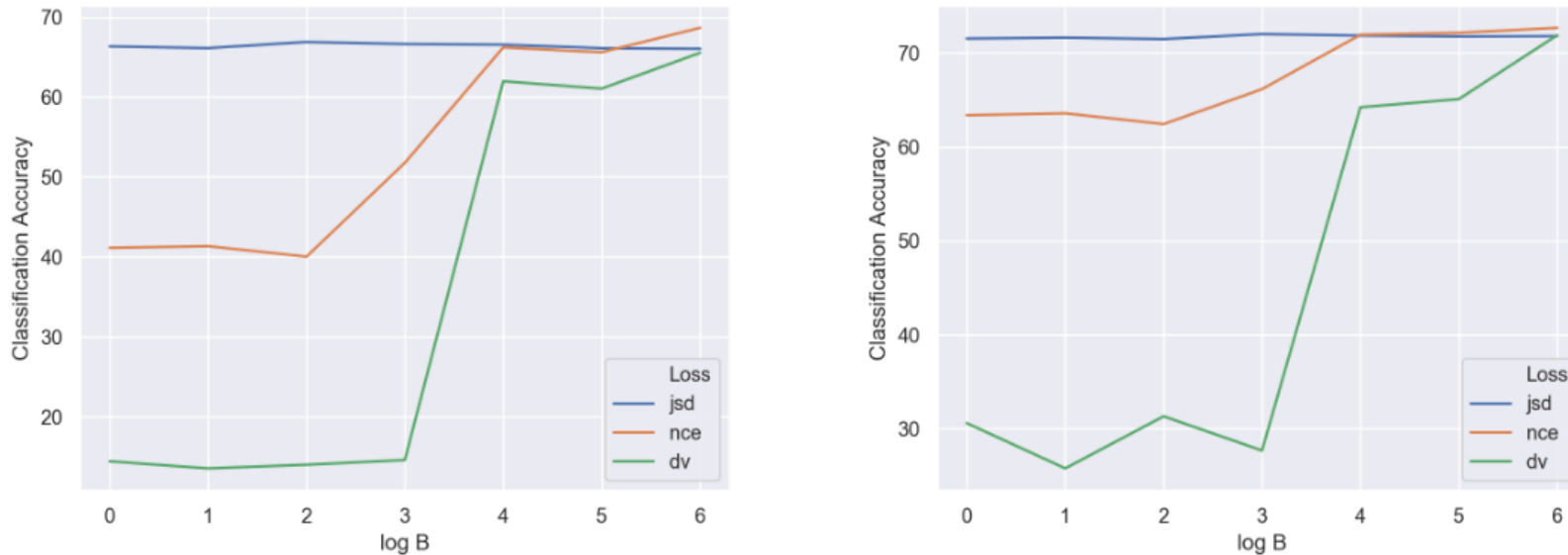


Figure 9: Classification accuracies (left: global representation, Y , right: convolutional layer) for CIFAR10, first training DIM, then training a classifier for 1000 epochs, keeping the encoder fixed.

Matching representations to a prior distribution

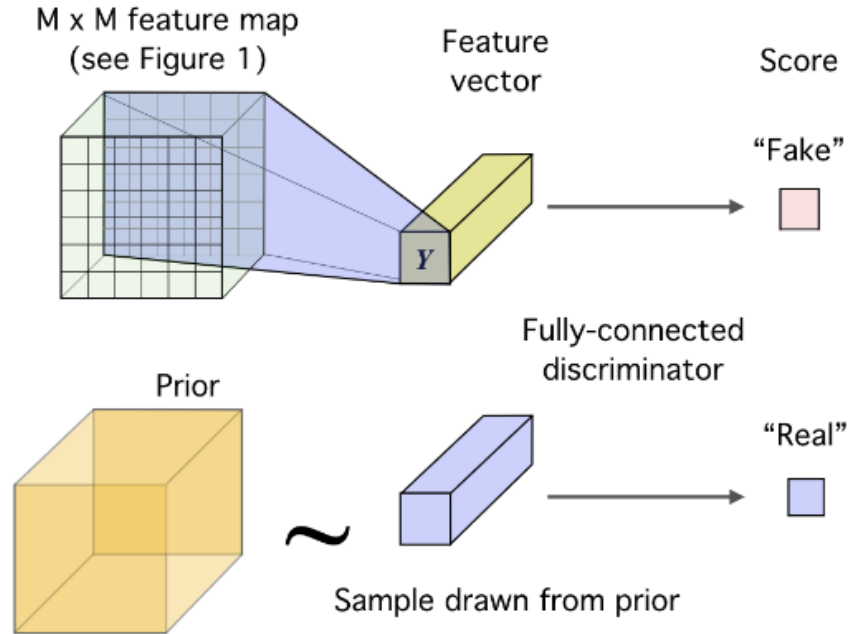


Figure 7: **Matching the output of the encoder to a prior.** “Real” samples are drawn from a prior while “fake” samples from the encoder output are sent to a discriminator. The discriminator is trained to distinguish between (classify) these sets of samples. The encoder is trained to “fool” the discriminator.

- DIM imposes statistical constraints onto learned representations by implicitly training the encoder so that the push-forward distribution, $\mathbb{U}_{\psi, \mathbb{P}}$, matches a prior, \mathbb{V} .
- This is done by training a discriminator, $D_{\phi} : \mathcal{Y} \rightarrow \mathbb{R}$, to estimate the divergence, $\mathcal{D}(\mathbb{V} \parallel \mathbb{U}_{\psi, \mathbb{P}})$:

$$(\hat{\omega}, \hat{\psi})_P = \underset{\psi}{\operatorname{argmin}} \underset{\phi}{\operatorname{argmax}} \mathbb{E}_{\mathbb{V}}[\log D_{\phi}(y)] + \mathbb{E}_{\mathbb{P}} \left[\log \left(1 - D_{\phi} \left(E_{\psi}(x) \right) \right) \right]$$

Idea

Summary)

- DIM sets the noise distribution to the product of marginal over X/Y , and the data distribution to the true joint.
- In practice, implementations of these estimators appear quite similar and can reuse most of the same code.
- InfoNCE and DV require a large number of negative samples (drawn from $\tilde{\mathbb{P}}$) to be competitive.
- We generate negative samples using all combinations of global and local features at all locations of the relevant feature map, across all images in a batch.
- (DIM local framework) For a batch of size B , that gives $O(B \times M^2)$ negative samples per positive example, which quickly becomes cumbersome with increasing batch size.

Idea

Combining all together

$$\operatorname{argmax}_{\omega_1, \omega_2, \psi} \left(\underbrace{\alpha \hat{I}_{\omega_1, \psi}(X; E_{\psi}(X))}_{\text{DIM (Global)}} + \underbrace{\frac{\beta}{M^2} \sum_{i=1}^{M^2} \hat{I}_{\omega_2, \psi} \left(C_{\psi}^{(i)}(X); E_{\psi}(X) \right)}_{\text{DIM (Local)}} + \operatorname{argmax}_{\psi} \operatorname{argmax}_{\phi} \gamma \hat{D}_{\phi}(\mathbb{V} || \mathbb{U}_{\psi, \mathbb{P}}) \right)$$

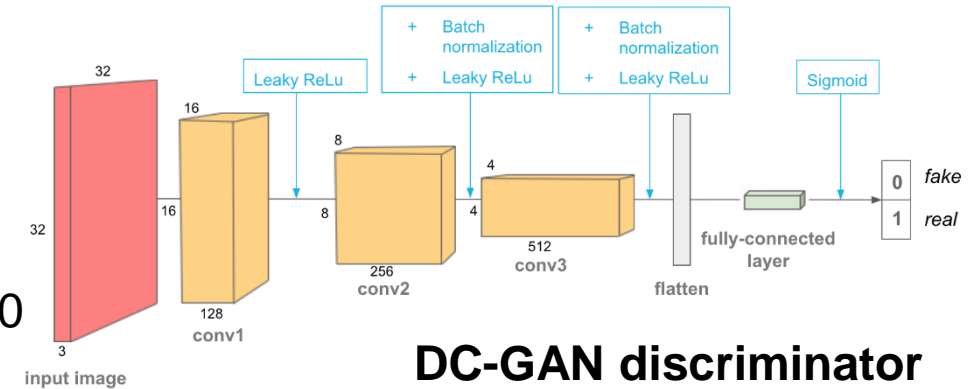
Prior matching

- ω_1, ω_2 : the discriminator parameters for the global and local objectives
- α, β, γ : hyper-parameters
- DIM(G) : $\alpha = 1, \beta = 0, \gamma = 1$
- DIM(L) : $\alpha = 0, \beta = 1, \gamma = 0.1$
- Prior distribution, $\mathbb{V} = [0,1]^{64}$, which worked better in practice than other priors, such as Gaussian, unit ball, or unit sphere.

Experiments

Experimental setting

- Encoder architecture :
 - similar to a DC-GAN discriminator for CIFAR-10 and CIFAR-100
 - AlexNet architecture for other datasets, Tiny ImageNet and STL-10



- For the DCGAN architecture, a **single hidden layer (fc)** with 1024 units was used after **the final convolutional layer (conv)**, and for the AlexNet architecture it was two hidden layers with 4096.
- For all experiments, **the output (Y)** of all encoders was a 64-dimensional vector.
- For all classification tasks, we built separate classifiers on **the high-level vector representation (Y)**, the **output of the previous fully-connected layer (fc)** and **the last convolutional layer (conv)**.

Experiments

Representation learning comparison across models

Table 1: Classification accuracy (top 1) results on CIFAR10 and CIFAR100. DIM(L) (i.e., with the local-only objective) outperforms all other unsupervised methods presented by a wide margin. In addition, DIM(L) approaches or even surpasses a fully-supervised classifier with similar architecture. DIM with the global-only objective is competitive with some models across tasks, but falls short when compared to generative models and DIM(L) on CIFAR100. Fully-supervised classification results are provided for comparison.

Model	CIFAR10			CIFAR100		
	conv	fc (1024)	Y(64)	conv	fc (1024)	Y(64)
Fully supervised	75.39			42.27		
VAE	60.71	60.54	54.61	37.21	34.05	24.22
AE	62.19	55.78	54.47	31.50	23.89	27.44
β -VAE	62.4	57.89	55.43	32.28	26.89	28.96
AAE	59.44	57.19	52.81	36.22	33.38	23.25
BiGAN	62.57	62.74	52.54	37.59	33.34	21.49
NAT	56.19	51.29	31.16	29.18	24.57	9.72
DIM(G)	52.2	52.84	43.17	27.68	24.35	19.98
DIM(L) (DV)	72.66	70.60	64.71	48.52	44.44	39.27
DIM(L) (JSD)	73.25	73.62	66.96	48.13	45.92	39.60
DIM(L) (infoNCE)	75.21	75.57	69.13	49.74	47.72	41.61

- Using a single hidden layer NN (200 units) with dropout, without fine-tuning the encoder.

Experiments

Representation learning comparison across models

Table 2: Classification accuracy (top 1) results on Tiny ImageNet and STL-10. For Tiny ImageNet, DIM with the local objective outperforms all other models presented by a large margin, and approaches accuracy of a fully-supervised classifier similar to the Alexnet architecture used here.

	Tiny ImageNet			STL-10 (random crop pretraining)			
	conv	fc (4096)	Y(64)	conv	fc (4096)	Y(64)	SS
Fully supervised	36.60			68.7			
VAE	18.63	16.88	11.93	58.27	56.72	46.47	68.65
AE	19.07	16.39	11.82	58.19	55.57	46.82	70.29
β -VAE	19.29	16.77	12.43	57.15	55.14	46.87	70.53
AAE	18.04	17.27	11.49	59.54	54.47	43.89	64.15
BiGAN	24.38	20.21	13.06	71.53	67.18	58.48	74.77
NAT	13.70	11.62	1.20	64.32	61.43	48.84	70.75
DIM(G)	11.32	6.34	4.95	42.03	30.82	28.09	51.36
DIM(L) (DV)	30.35	29.51	28.18	69.15	63.81	61.92	71.22
DIM(L) (JSD)	33.54	36.88	31.66	72.86	70.85	65.93	76.96
DIM(L) (infoNCE)	34.21	38.09	33.33	72.57	70.00	67.08	76.81

- Semi-supervised learning (SS) for STL-10 : fine-tuning the complete encoder by adding a small neural network on top of the last convolutional layer (matching architectures with a standard fully-supervised classifier).
- STL-10 : 10 classes, 500 training and 800 test images per class, 100,000 unlabeled images.

Experiments

Representation learning comparison across models

Table 4: Extended comparisons on CIFAR10. Linear classification results using SVM are over five runs. MS-SSIM is estimated by training a separate decoder using the fixed representation as input and minimizing the L_2 loss with the original input. Mutual information estimates were done using MINE and the neural dependence measure (NDM) were trained using a discriminator between unshuffled and shuffled representations.

Model	Proxies				Neural Estimators	
	SVM (conv)	SVM (fc)	SVM (Y)	MS-SSIM	$\hat{I}_\rho(X, Y)$	NDM
VAE	53.83 ± 0.62	42.14 ± 3.69	39.59 ± 0.01	0.72	93.02	1.62
AAE	55.22 ± 0.06	43.34 ± 1.10	37.76 ± 0.18	0.67	87.48	0.03
BiGAN	56.40 ± 1.12	38.42 ± 6.86	44.90 ± 0.13	0.46	37.69	24.49
NAT	48.62 ± 0.02	42.63 ± 3.69	39.59 ± 0.01	0.29	6.04	0.02
DIM(G)	46.8 ± 2.29	28.79 ± 7.29	29.08 ± 0.24	0.49	49.63	9.96
DIM(L+G)	57.55 ± 1.442	45.56 ± 4.18	18.63 ± 4.79	0.53	101.65	22.89
DIM(L)	63.25 ± 0.86	54.06 ± 3.6	49.62 ± 0.3	0.37	45.09	9.18

$\alpha = 0.5 \quad \beta = 0.1 \rightarrow$

- (linear-) SVM : to test the linear separability of learned representations
- MS-SSIM : using a decoder trained on the L_2 reconstruction loss.
- $\hat{I}_\rho(X, E_\psi(X))$: mutual information between the input and the representation estimated by MINE (DV estimator).
- Neural Dependency Measure : KLD between $E_\psi(X)$ and shuffled $E_\psi(X)$.

Experiments

Representation learning comparison across models

Table 5: Augmenting infoNCE DIM with additional structural information – adding coordinate prediction tasks or occluding input patches when computing the global feature vector in DIM can improve the classification accuracy, particularly with the highly-compressed global features.

Model	CIFAR10			CIFAR100		
	Y (64)	fc (1024)	conv	Y (64)	fc (1024)	conv
DIM	70.65	73.33	77.46	44.27	47.96	49.90
DIM (coord)	71.56	73.89	77.28	45.37	48.61	50.27
DIM (occlude)	72.87	74.45	76.77	44.89	47.65	48.87
DIM (coord + occlude)	73.99	75.15	77.27	45.96	48.00	48.72

- For occlusions, we randomly occlude part of the input when computing the global features, but compute local features using the full input by maximizing $\mathbb{E} \left[\log p_{\theta} \left((i, j) | y, c_{(i, j)} \right) \right]$.
- It maximizes the model's ability to predict the coordinates (i, j) of a local feature $c_{(i, j)} = C_{\psi}^{(i, j)}(x)$ after computing the global features $y = E_{\psi}(x)$.
- We can extend the task to minimize the conditional MI given global features y between pairs of local features $(c_{(i, j)}, c_{(i', j')})$ and their relative coordinates $(i - i', j - j')$, by maximizing $\mathbb{E} \left[\log p_{\theta} \left((i - i', j - j') | y, c_{(i, j)}, c_{(i', j')} \right) \right]$.

Experiments

Representation learning comparison across models

Table 5: Augmenting infoNCE DIM with additional structural information – adding coordinate prediction tasks or occluding input patches when computing the global feature vector in DIM can improve the classification accuracy, particularly with the highly-compressed global features.

Model	CIFAR10			CIFAR100		
	Y (64)	fc (1024)	conv	Y (64)	fc (1024)	conv
DIM	70.65	73.33	77.46	44.27	47.96	49.90
DIM (coord)	71.56	73.89	77.28	45.37	48.61	50.27
DIM (occlude)	72.87	74.45	76.77	44.89	47.65	48.87
DIM (coord + occlude)	73.99	75.15	77.27	45.96	48.00	48.72

- Can be interpreted as generalizations of inpainting and context prediction tasks, which have previously been proposed for self-supervised learning
→ similar to the contrastive predictive coding (Aaron v. d. Oord et. al., 2018, <https://arxiv.org/abs/1807.03748>).
- Augmenting DIM with these tasks helps move our method further towards learning representations which encode images not just in terms of compressing their low-level (e.g. pixel) content, but in terms of distributions over relations among high-level features extracted from their low-level content.

Conclusion

- DIM performs Unsupervised representation learning by maximizing mutual information.
- The authors used the variational MI estimators, DV/JS-div/infoNCE, which enables maximizing the (estimated-) MI by sampling instances from the joint and the product of marginals.
- Also, well-designed tasks, such as
 - MI maximization between the input and the global features → Global DIM
 - MI maximization between the local and the global feature → Local-DIM
 - conditional MI maximization by predicting occlusions and relative coordinatescan improve the quality of representations and good performances on down-stream tasks (classifications).

Notes and Other literatures

- Task designing is important.
 - g_1 and g_2 in $\max_{g_1 \in \mathcal{G}_1, g_2 \in \mathcal{G}_2} \hat{I}(g_1(X^{(1)}); g_2(X^{(2)}))$ significantly affects the quality of learned representations.
- Theoretical analysis and critics have been investigated in “On Mutual Information Maximization for Representation Learning, <https://arxiv.org/abs/1907.13625>”.
- Choice of an MI-estimator $\hat{I}(X; Y)$ (variational bound of true MI) also affects the representation learning
 - Wasserstein Dependency Measure for Representation Learning, <https://arxiv.org/abs/1903.11780>
 - On Variational Bounds of Mutual Information, <https://arxiv.org/abs/1905.06922>
 - Understanding the Limitations of Variational Mutual Information Estimators, <https://arxiv.org/abs/1910.06222>