Information theory in deep learning

Presenter: Zhe Wang

https://qdata.github.io/deep2Read

Zhe Wang

201909

Content

Mutual Information

For two variables X, Z, their mutual information(MI) is defined as:

$$MI(X,Z) = KL(P(X,Z)||P(X)P(Z)) = \int p(x,z) \log \frac{p(x,z)}{p(x)p(z)} dxdz \quad (1)$$

Properties

- $MI(X,Z) \geqslant 0$,
- $MI(X,Z) = 0 \Longleftrightarrow X, Z$ are independent,
- $MI(X,Z) = MI(Z,X)$
- $MI(X,Z) = H(X) - H(X|Z) = H(Z) - H(Z|X)$

$MI(X,Z)$ measures how much we can learn about X from Y. It is intractable for the unknown joint and marginal distribution.

Mutual information neural estimation[1]

Goal: Estimate mutual information from data.

Main idea: Parameterize the lower bound of mutual information and recover the mutual information by maximizing the lower bound.

The KL divergence admits the following dual representation

$$KL(P||Q) \geqslant \sup_{T:\Omega \to R} E_P(T) - \log(E_Q[e^T])$$

[1] Mohamed Ishmael Belghazi et.al., MILA, ICML2018

Use the neural network to model the function T, and use the empirical value as an approximation of expectation.

$$MI(X,Z) = \sup_{\theta} \frac{1}{N} \sum_{i=1}^{N} T_\theta([x,z]_i) - \log(\frac{1}{M} e^{T_\theta(x_k,z_k)}),$$

where $[x,z]_i$ are sampled from $P(X,Z)$, and (x_k,z_k) are sampled from $P(X)P(Z)$.

Theoretical analysis

Theorem

Let $\epsilon > 0$. There exists a neural network parametrizing functions T_θ with parameters θ in some compact domain $\theta \in R^k$, such that:

$$||MI(X, Z) - MI_\theta(X, Z)|| \leqslant \epsilon$$

Theorem

Given a family of function with enough capacity and large dataset, with universal approximation ability of neural network, empirical value can be arbitrarily close to expected value.

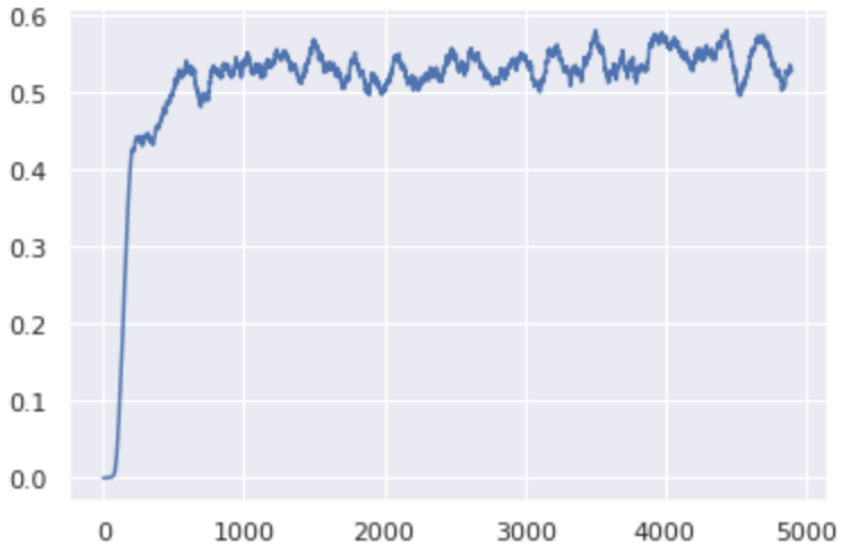
$$||MI_\theta(X, Z) - MI_\theta(X_i, Z_i)|| \leqslant \epsilon$$

Experiments

Simulation dataset: $(X, Z) \sim N(\mu, \Sigma)$, where $\mu = [\mu_x, \mu_z]$, and $\Sigma = \begin{bmatrix} \Sigma_{x,x}, \Sigma_{x,z} \\ \Sigma_{z,x}, \Sigma_{z,z} \end{bmatrix}$

For normal distribution, the analytic form of KL divergence is tractable:

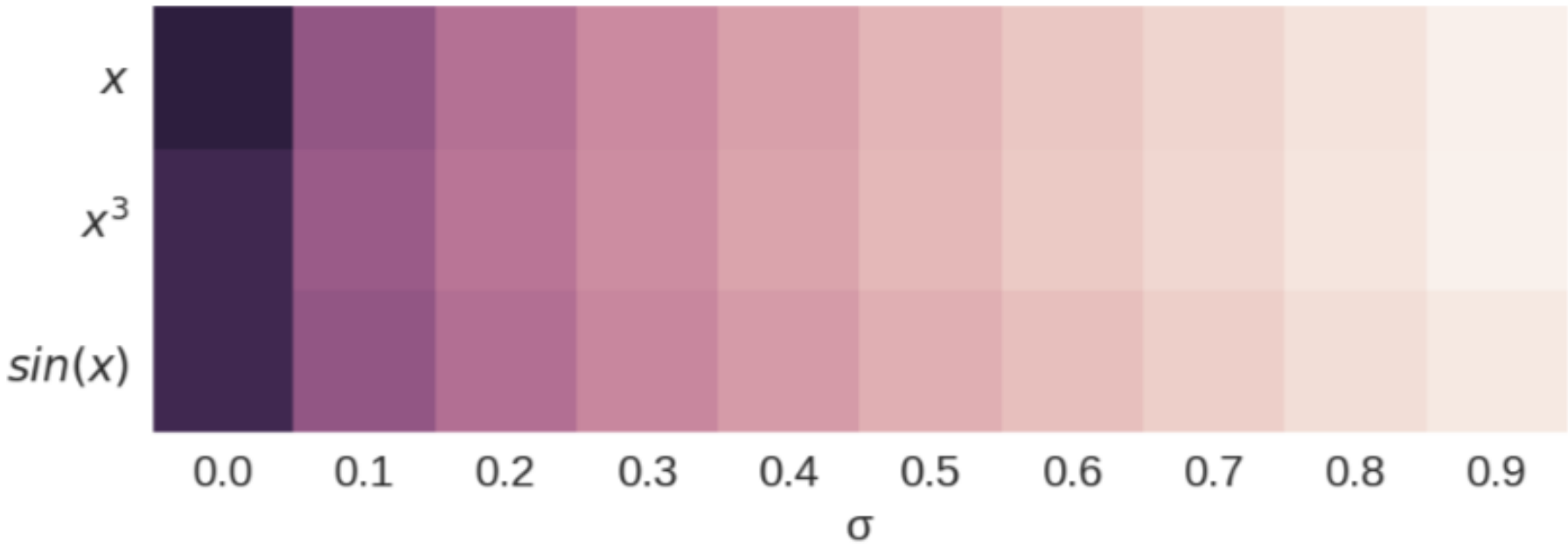
$$KL(p||q) = \frac{1}{2}[\log \frac{|\Sigma_2|}{|\Sigma_1|} - n + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_1^T \Sigma_2^{-1} \mu_1 - 2\mu_2^T \Sigma_2^{-1} \mu_1 + \mu_2^T \Sigma_2^{-1} \mu_2)]$$



Invariant Property

Suppose $Y = f(X) + \sigma \cdot \epsilon$, where f is a deterministic function, then once $\sigma \cdot \epsilon$ is fixed, $MI(X, Y)$ remains invariant.

Experiment result:

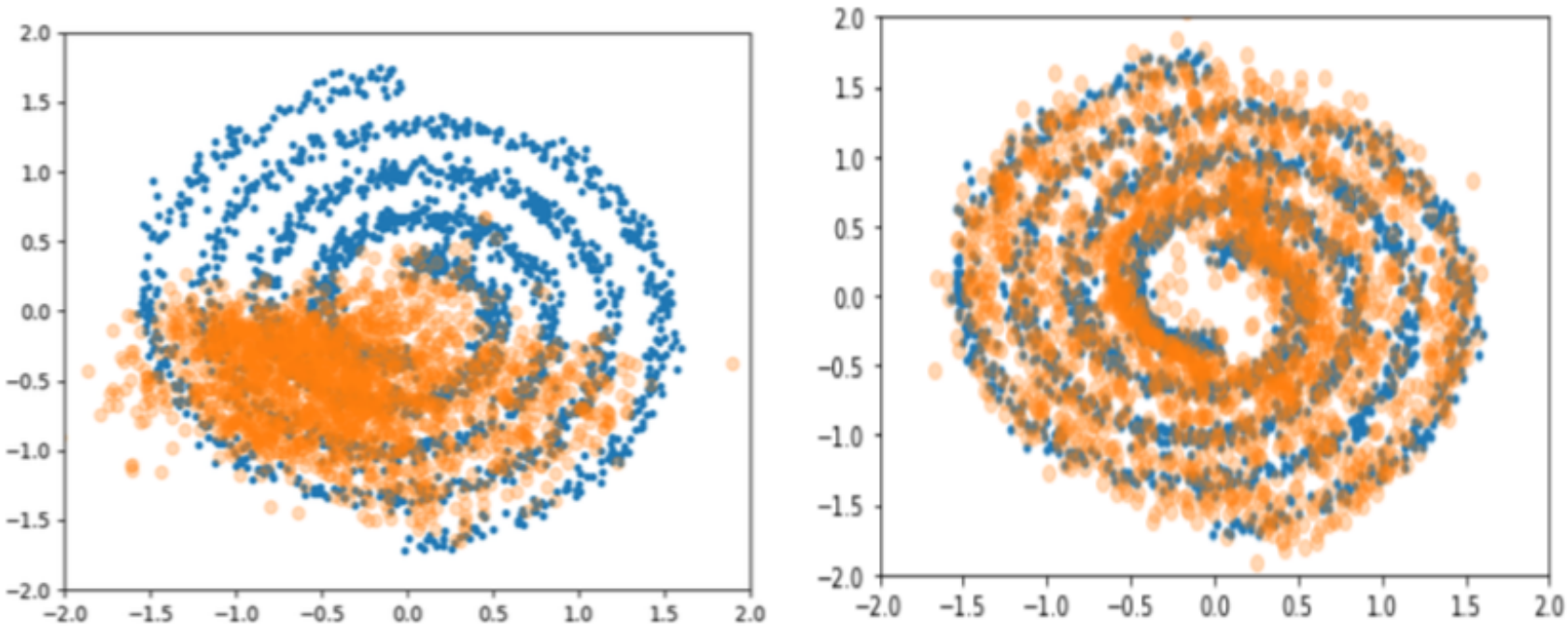


MINE can effectively palliate model collapse in GANs.
Vanilla GAN:

$$\min_G \max_D \log_{x\sim P_r} D(x) + \log_{z\sim N}(1 - D(G(z)))$$

New regularization for G, the mutual information between the samples and the codes:

$$I(G[\epsilon, C], C)$$



(a) GAN (b) GAN+MINE

(a) Original data (b) GAN (c) GAN+MINE

Figure 4. Kernel density estimate (KDE) plots for GAN+MINE samples and GAN samples on 25 Gaussians dataset.

Information Bottleneck

Information Bottleneck(IB):
learn a representation that an input $x \in X$ contains about an output $y \in Y$.
An optimal rep z would capture the relevant features of X, while diminish the irrelevant parts which do not contribute to the prediction of Y
the optimization objective:

$$\min_{Z} H(Y|Z) + \beta I(X, Z)$$

Content

Representation Learning

What is a good representation?

a good representation is often one that captures the posterior distribution of the underlying explanatory factors for the observed input (Bengio et.al., 2013)

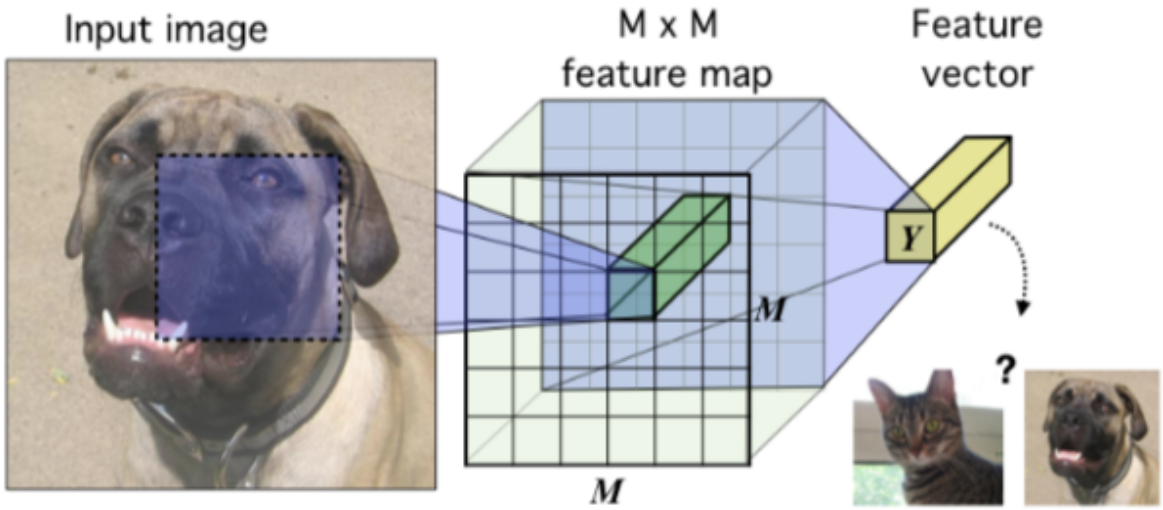
- high mutual information with the input(not low-level noise)
- task-dependent statistical properties(independent, separable)
- structure contained (high-level semantic information)

Deep InfoMax[2]

Main idea: Maximize the mutual information between the representations and input.

In most benchmark deep models, the computational graph contains several stages:

$$X \rightarrow C_\psi(X) \rightarrow E_\psi(X) = f_\psi(C_\psi(X)) \rightarrow \textit{classifier}$$

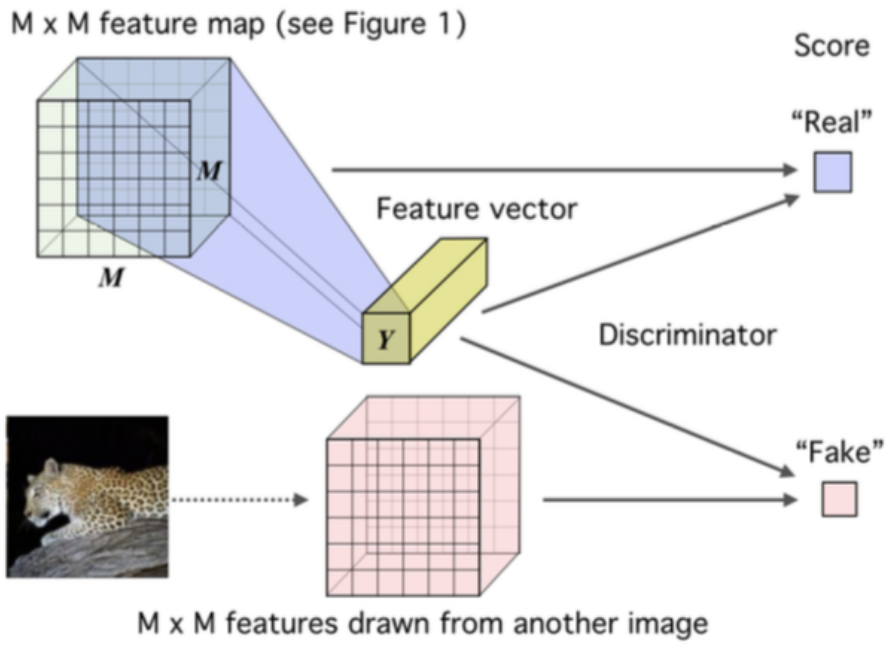


[2]R Devon Hjelm, et.al, MILA, ICLR 2019

Model

Goal: maximize the mutual information $MI(P_X, P_{E_\Psi(X)})$.
In order to optimize the goal, we need to collect data from joint distribution and marginal distribution.

- $X, E_\Psi(X)$ is sampled from the joint distribution
- $X, E_\Psi(\hat{X})$ is sampled from the product of marginal distributions.



In order to estimate the mutual information, we have several strategies:

- Use MINE:

$$MI_w(X, E_\Psi(X)) = \sup_{T_w:\Omega\to R} E_{P(X,E_\Psi(X))}(T_w) - \log(E_{P(X)\times P(E_\Psi(X))}[e^{T_w}])$$

- Use other divergence

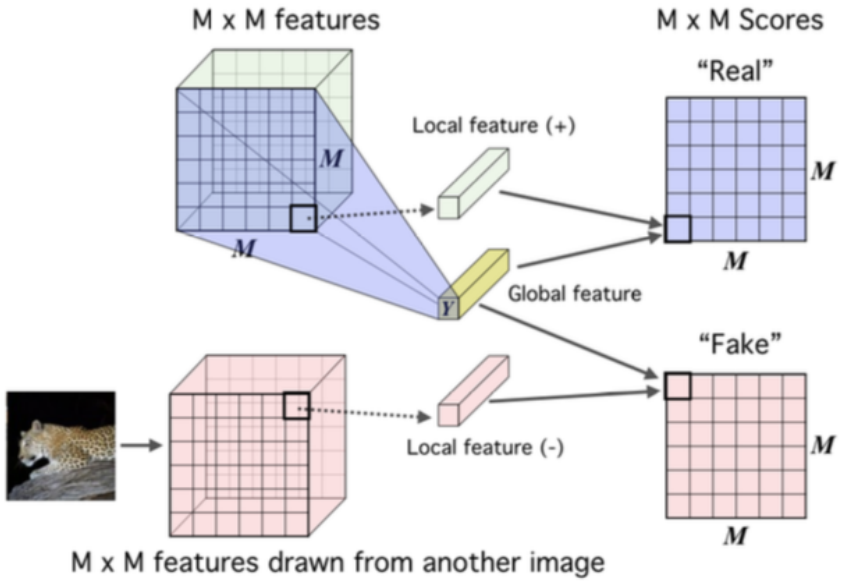
$$D^{JSD}(X, E_\Psi(X)) = \sup_{T_w:\Omega\to R} E_{P(X,E_\Psi(X))}(-\sigma(-T_w)) - E_{P(X)\times P(E_\Psi(X))}[\sigma(T_w)]$$

- Use the Noise-Contrastive Estimation(NCE):

$$I^{NCE}(X, E_\Psi(X)) = E_{P(x)}(T_w(x, E_\Psi(x)) - E_{P(X)}[\log \sum_{x'} e^{T_w(x', E_\Psi(x)}])$$

Local variants

Motivation:
Maximize the average MI between the high-level representation and local patches of the image.



$$MI_{local} = \frac{1}{M^2} \sum_{i=1}^{M^2} I(C_{\psi}^{i}(x), E_{\psi}(x))$$

Prior Matching

If we also have domain knowledge about the representation, we would like to match the prior distribution, the loss is defined as a GANs loss:

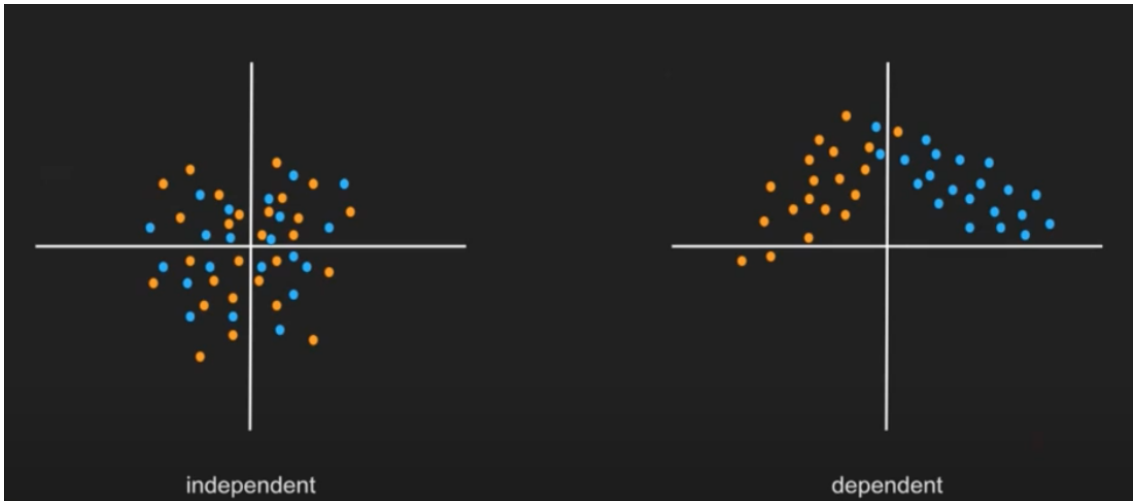
$$\min_{E_\psi} \max_{D} \log_{x \sim P_r} D(x) + \log_{z \sim P}(1 - D(E_\psi(z)))$$

The final loss is the linear combination of three losses (global loss, local loss and prior matching loss).

Evaluation

How to measure the quality of learned representations?

- Use the representations as the input of classifiers(SVM and NN), and compare the accuracy.
- Calculate the mutual information between the input and representations.
- Add a decoder to reconstruct the input with l2 loss
- Measure the independence of the representation using a discriminator(NDM).



Results

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**

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 $L2$ 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	$\widehat{I}_\rho(X,Y)$	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