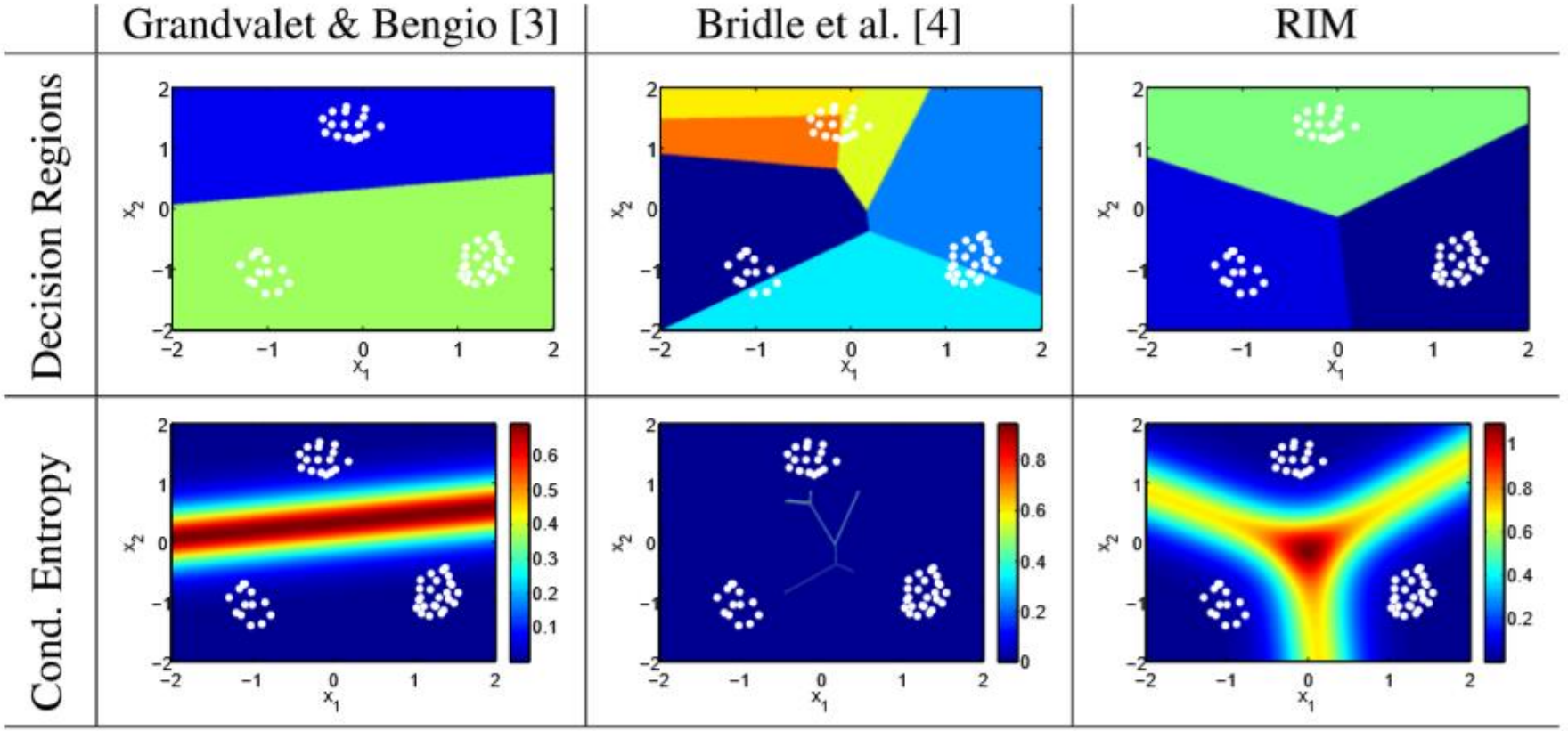
Discriminative Clustering by Regularized Information Maximization(RIM)-NIPS2010



Method	Conditional Entropy	Mutual Information	Mutual Information + complex decision boundaries penalty
Disadvantage	Conditional entropy may be reduced by simply removing decision boundaries	Each data point x_i into its own category y_i	-

Goals

1. Decision boundaries should not be located in regions of the input space that are densely populated with datapoints
2. Datapoints should be classified with large margin.
3. We prefer configurations in which category labels are *assigned evenly*.
   Why? Because conditional entropy may be reduced by simply removing decision boundaries(Refer slide 2)

Objective function

Penalizes conditional models with complex decision boundaries

$$R(\mathbf{W};\lambda) = \lambda \sum_k \mathbf{w}_k^T \mathbf{w}_k,$$

max $F(\mathbf{W};\mathbf{X},\lambda) = I_{\mathbf{W}}\{y;\mathbf{x}\} - R(\mathbf{W};\lambda)$ (2)

Where,

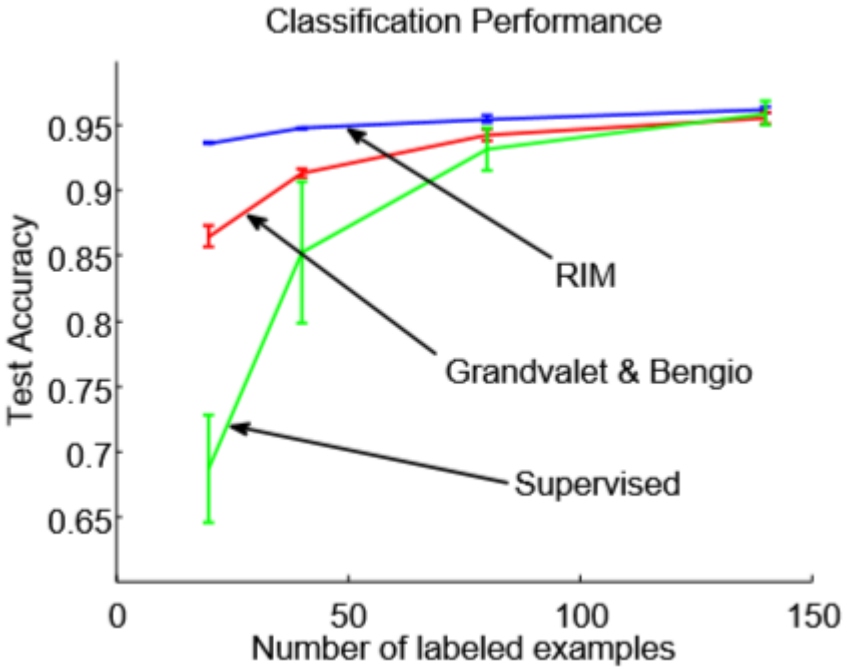
$$I_{\mathbf{W}}\{y;\mathbf{x}\} = H\{\hat{p}(y;\mathbf{W})\} - \frac{1}{N}\sum_i H\{p(y|\mathbf{x}_i,\mathbf{W})\} \quad (1)$$

Goal 3

Goal 1

Semi-supervised Setting

- Objective function: $S(\mathbf{W};\tau,\lambda) = \tau I_{\mathbf{W}}\{y;\mathbf{x}\} - R(\mathbf{W};\lambda) + \sum_i \log p(y_i|\mathbf{x}_i^L,\mathbf{W})$
- τ is the tradeoff between labeled and unlabeled examples.



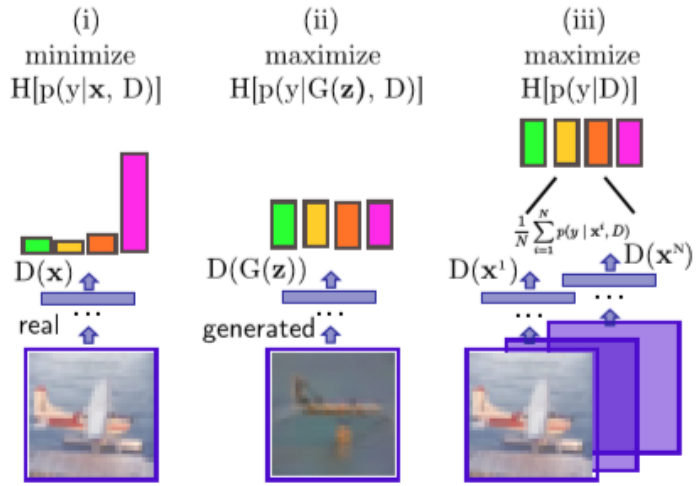
Semi-supervised learning on Caltech Images

Conclusion: This suggests that ***incorporating prior knowledge*** about class size distributions (in this case, we use a uniform prior) may be useful in semi-supervised learning.

UNSUPERVISED AND SEMI-SUPERVISED LEARNING WITH CATEGORICAL GENERATIVE ADVERSARIAL NETWORKS (CatGAN)

An Extension to RIM

Goals



Discriminator

1. Be **certain** of prediction from **true** data
2. Be **uncertain** of prediction from generated **fake** samples
3. Use all classes **equally** (uniform prior $P(y)$ over classes)

Generator

1. Generate fake samples with **certain** class assignments
2. Generate fake samples with **equal** distribution over all classes

Objective function:

D3 D1 D2 Supervised loss

$$\mathcal{L}_D = \max_D H_{\mathcal{X}}\left[p(y \mid D)\right] - \mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\left[H\left[p(y \mid \mathbf{x}, D)\right]\right] + \mathbb{E}_{\mathbf{z}\sim P(\mathbf{z})}\left[H\left[p(y \mid G(\mathbf{z}), D)\right]\right] + \lambda \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{X}^L}\left[CE\left[\mathbf{y}, p(y \mid \mathbf{x}, D)\right]\right]$$

$$\mathcal{L}_G = \min_G -H_G\left[p(y \mid D)\right] + \mathbb{E}_{\mathbf{z}\sim P(\mathbf{z})}\left[H\left[p(y \mid G(\mathbf{z}), D)\right]\right],$$

G2 G1

Mutual Information

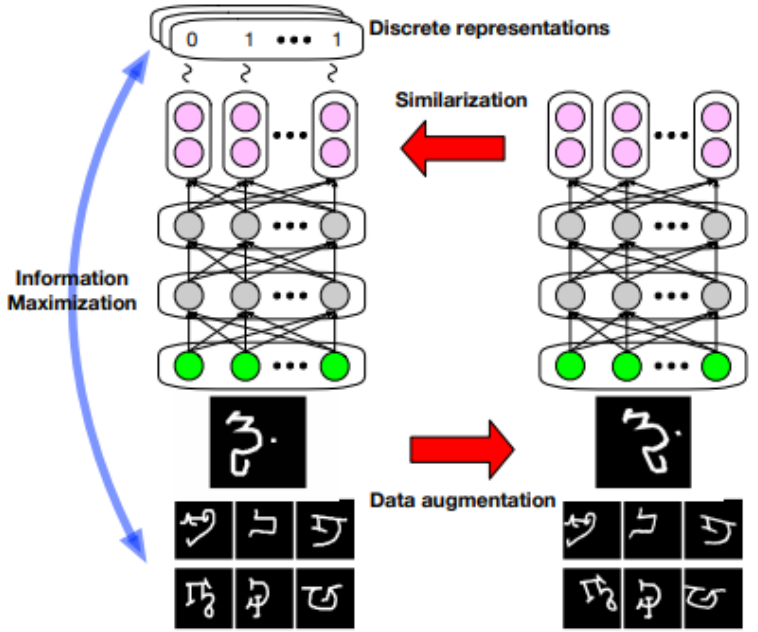
Learning Discrete Representations via Information Maximizing Self-Augmented Training-ICML2017

Objectives: Learning discrete representations of data

(obtain a function $f: X \rightarrow Y$, that maps similar inputs into similar discrete representations.)

Two techniques

1. <u>Data augmentation</u> to impose the invariance on discrete representations.
2. <u>Maximize the information theoretic dependency</u> between data(input) and their predicted(output) discrete representations.



Objective function

$$\mathcal{R}(\theta) - \lambda \boldsymbol{I}(X;Y), \tag{1}$$

$$\mathcal{R}_{\mathrm{SAT}}(\theta;T) - \lambda\left[H(Y) - H(Y|X)\right], \tag{7}$$

$$\min_{\theta} \ \mathcal{R}_{\mathrm{SAT}}(\theta;T) + \lambda H(Y|X),$$
$$\text{subject to } \ \mathrm{KL}[p_\theta(y)||\, q(y)] \leq \delta, \tag{10}$$

where $H(Y) = \log K - \mathrm{KL}[p_\theta(y)||\, q(y)]$

Predicted distribution

Class prior

Results

Table 3. Comparison of clustering accuracy on eight benchmark datasets (%). Averages and standard deviations over twelve trials were reported. Results marked with † were excerpted from Xie et al. (2016).

Method	MNIST	Omniglot	STL	CIFAR10	CIFAR100	SVHN	Reuters	20news
K-means	53.2	12.0	85.6	34.4	21.5	17.9	54.1	15.5
dAE+K-means	79.8 †	14.1	72.2	44.2	20.8	17.4	67.2	22.1
DEC	84.3 †	5.7 (0.3)	78.1 (0.1)	**46.9 (0.9)**	14.3 (0.6)	11.9 (0.4)	67.3 (0.2)	30.8 (1.8)
Linear RIM	59.6 (2.3)	11.1 (0.2)	73.5 (6.5)	40.3 (2.1)	23.7 (0.8)	20.2 (1.4)	62.8 (7.8)	**50.9 (3.1)**
Deep RIM	58.5 (3.5)	5.8 (2.2)	92.5 (2.2)	40.3 (3.5)	13.4 (1.2)	26.8 (3.2)	62.3 (3.9)	25.1 (2.8)
Linear IMSAT (VAT)	61.1 (1.9)	12.3 (0.2)	91.7 (0.5)	40.7 (0.6)	23.9 (0.4)	18.2 (1.9)	42.9 (0.8)	43.9 (3.3)
IMSAT (RPT)	89.6 (5.4)	16.4 (3.1)	92.8 (2.5)	45.5 (2.9)	24.7 (0.5)	35.9 (4.3)	**71.9 (6.5)**	24.4 (4.7)
IMSAT (VAT)	**98.4 (0.4)**	**24.0 (0.9)**	**94.1 (0.4)**	45.6 (0.8)	**27.5 (0.4)**	**57.3 (3.9)**	**71.0 (4.9)**	31.1 (1.9)

Generative Adversarial Image Synthesis with Decision Tree Latent Controller(DTLC-GAN)

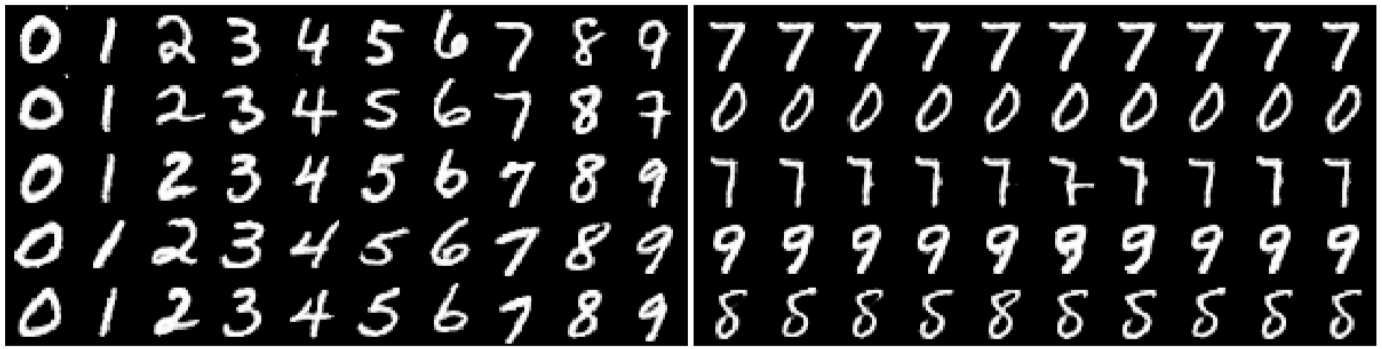
CVPR2018

An Extension to InfoGAN

[DTLC-GAN Demo]

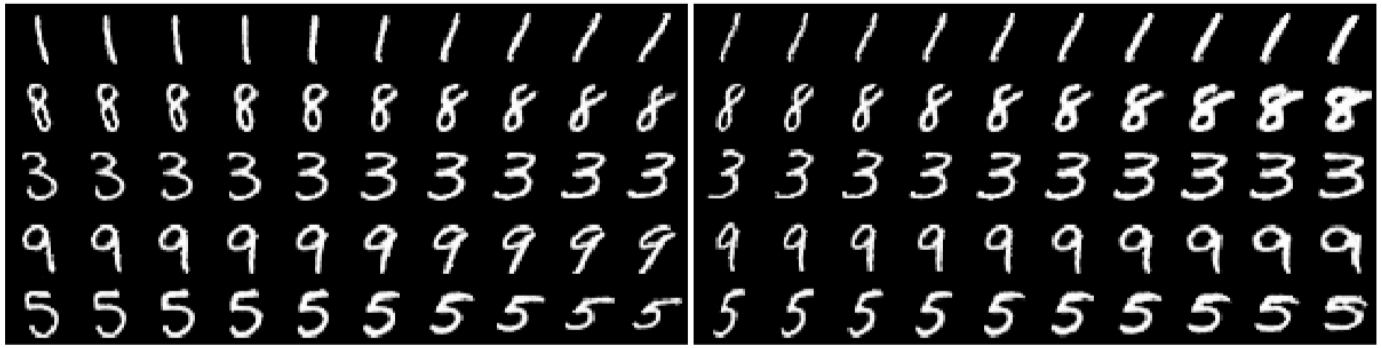
Recall InfoGAN

- Separate vector *z* into *z'* and *c*, varying value of particular dimension in vector *c* changes the image features



(a) Varying c_1 on InfoGAN (Digit type)

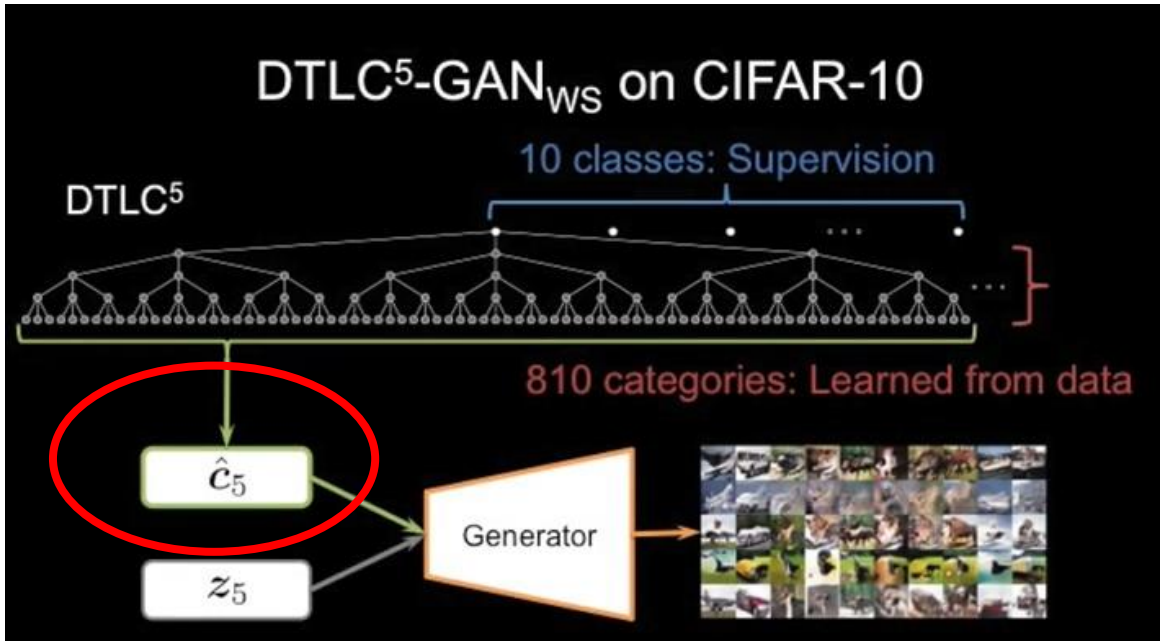
(b) Varying c_1 on regular GAN (No clear meaning)



(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)

(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

Network structure



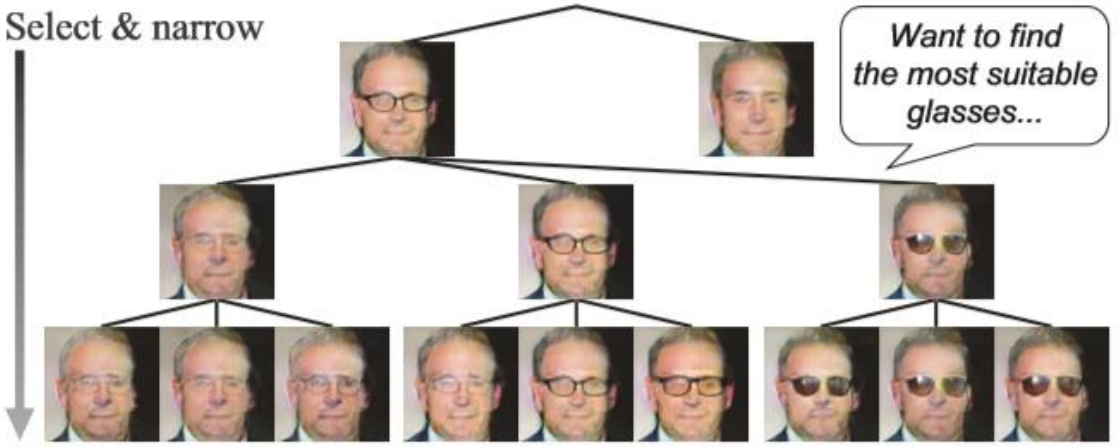


Figure 1. **Examples of image generation under control using DTLC-GAN:** DTLC-GAN enables image generation to be controlled in coarse-to-fine manner, i.e., "selected & narrowed." Our goal is to discover such hierarchically interpretable representations without relying on detailed supervision.

Aim: How to derive hierarchically interpretable representations in a deep generative model?

Sampling scheme

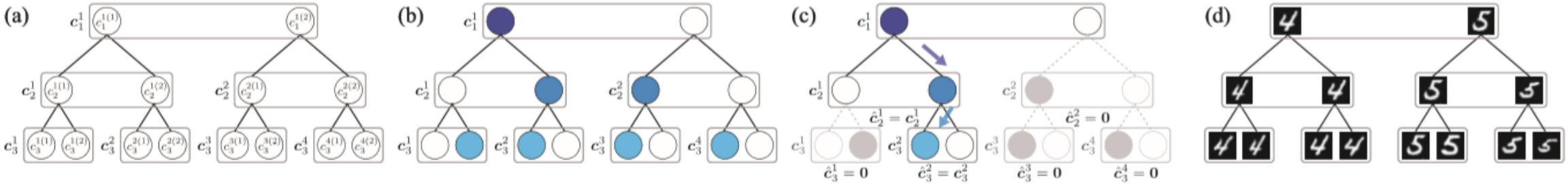
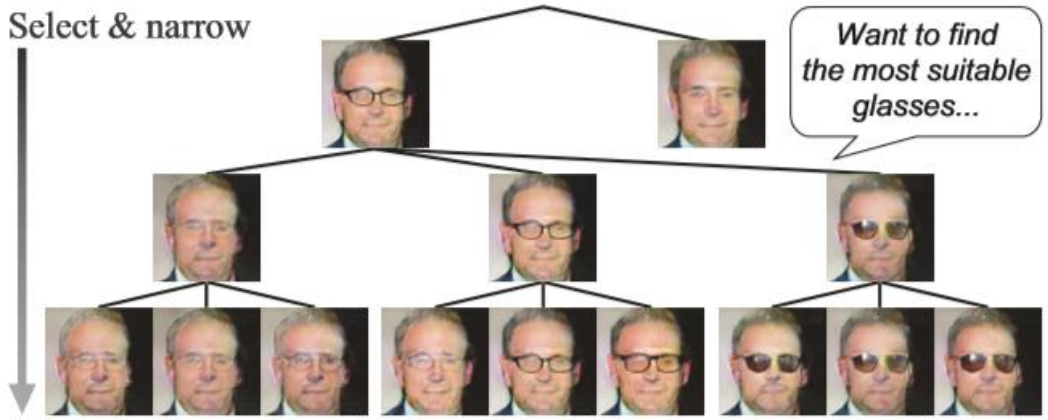


Figure 3. **Sampling example using three-layer DTLC:** (a) Architecture of three-layer DTLC where $k_1, k_2, k_3 = 2$. (b) Sampling example in Step 1. Each code is sampled from categorical distribution. Filled and open circles indicate 1 and 0, respectively. (c) Sampling example in Steps 2 and 3. ON or OFF of child node codes is selected by parent node codes. This execution is conducted recursively from highest layer to lowest layer. This imposes hierarchical inclusion constraints on sampling. (d) Sample images generated using this controller. Each image corresponds to each latent code. We tested on subset of MNIST dataset, which includes "4" and "5" digit images. This is relatively easy dataset; however, it is noteworthy that hierarchically disentangled representations, such as "4" or "5" in first layer and "narrow-width 4" or "wide-width 4" in second left layer, are learned in fully unsupervised manner.

- **Instability caused by random sampling** of the lower layer codes can degrade the learning performance.
- Humans learn much better when the examples are **not randomly** presented but **organized** in a meaningful order which illustrates **gradually** more complex ones.
- 遍历decision tree得latent code *c*

Disentanglement(纠缠)



1. Disentanglement between the control target (e.g., glasses) and unrelated factors (e.g., identity)
2. Coarse-to-fine disentanglement between layers
   - Higher layer codes capture rough features, lower layer codes capture detailed features
3. Inner-layer disentanglement to control semantic features independently
   - One captures thick glasses while the other code captures thin glasses

Solution: We proposed a **hierarchical conditional mutual information regularization** (HCMI)

Objective function

HCMI: Hierarchical Conditional Mutual Information Regularization
MI: Mutual Information
AC: Auxiliary Classifier

- Regularization for Second Layer to L-th layer:

$$\begin{aligned}&\mathcal{L}_{\mathrm{HCMI}}(G, Q_l^m)\\ =&\mathbb{E}_{\boldsymbol{c}\sim P(\boldsymbol{c}|\boldsymbol{p}),\boldsymbol{x}\sim G(\hat{\boldsymbol{c}}_L,\boldsymbol{z})}[\log Q_l^m(\boldsymbol{c}|\boldsymbol{x},\boldsymbol{p})].\end{aligned}$$

- Regularization for First Laver:

$$\mathcal{L}_{\mathrm{MI}}(G, Q_1) = \mathbb{E}_{\boldsymbol{c}_1\sim P(\boldsymbol{c}_1),\boldsymbol{x}\sim G(\hat{\boldsymbol{c}}_L,\boldsymbol{z})}[\log Q_1(\boldsymbol{c}_1|\boldsymbol{x})].$$

- In *weakly supervised* setting, auxiliary classifier regularization is used:

$$\begin{aligned}\mathcal{L}_{\mathrm{AC}}(G, Q_1) = &\mathbb{E}_{\boldsymbol{c}_1\sim P(\boldsymbol{c}_1),\boldsymbol{x}\sim G(\hat{\boldsymbol{c}}_L,\boldsymbol{z})}[\log Q_1(\boldsymbol{c}_1|\boldsymbol{x})]\\ &+ \mathbb{E}_{\boldsymbol{c}_1,\boldsymbol{x}\sim P_{\mathrm{data}}(\boldsymbol{c}_1,\boldsymbol{x})}[\log Q_1(\boldsymbol{c}_1|\boldsymbol{x})].\end{aligned}$$

- Full objectives:

$$\mathcal{L}_{\mathrm{Full}}(D, G, Q_1, \cdots, Q_L) = \mathcal{L}_{\mathrm{GAN}}(D, G) - \lambda_1\mathcal{L}_{\mathrm{MI/AC}}(G, Q_1) - \sum_{l=2}^{L}\lambda_l\mathcal{L}_{\mathrm{HCMI}}(G, Q_l).$$

Curriculum for Regularization

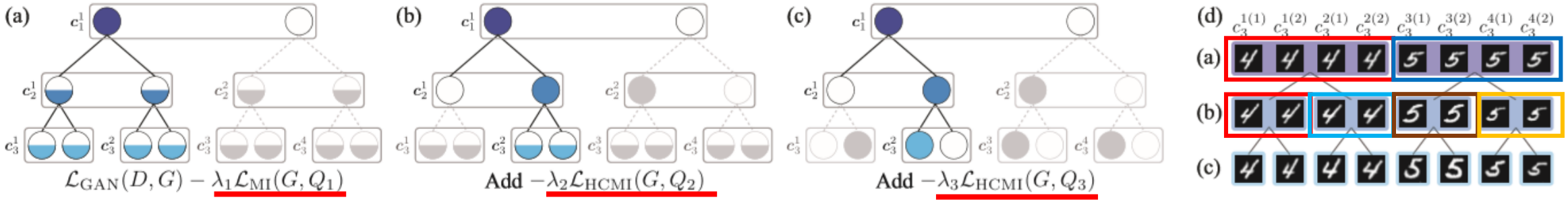


Figure 4. **Example of curriculum learning:** (a) We first learn disentangled representations in first layer. To do this, we only use regularization for this layer and fix and set average value to lower layer codes. (b)(c) We then learn disentangled representations in second and third layers in layer-by-layer manner. We add regularization and sampling in turn depending on training phase. (d) Image samples generated in each phase. In phase (a), first-layer codes are learned, while second- and third-layer codes are fixed; therefore, 2 disentangled representations are obtained. In phase (b), first- and second-layer codes are learned, while third-layer codes are fixed; therefore, 2×2 disentangled representations are obtained. In phase (c), all codes are learned; therefore, $2 \times 2 \times 2$ disentangled representations are obtained.

- Regularized from top to bottom
  i.e: Regularize first layer, fix and set average value for lower layer codes

Results

Model	Inception Score	Adversarial Accuracy	Adversarial Divergence
GAN	7.09 ± 0.09	-	-
AC-GAN	7.41 ± 0.06	50.99 ± 0.55	2.07 ± 0.02
DTLC^2-GAN$_{WS}$	7.39 ± 0.03	55.10 ± 0.48	$\mathbf{1.82 \pm 0.03}$
DTLC^3-GAN$_{WS}$	7.35 ± 0.09	55.20 ± 0.47	1.95 ± 0.05
DTLC^4-GAN$_{WS}$	7.46 ± 0.06	56.19 ± 0.36	1.93 ± 0.05
DTLC^5-GAN$_{WS}$	$\mathbf{7.51 \pm 0.06}$	$\mathbf{58.87 \pm 0.52}$	1.83 ± 0.04
Real Images	11.24 ± 0.12	85.77 ± 0.22	0
State-of-the-Art	8.59 ± 0.12 [13]	44.22 ± 0.08 [49]	5.57 ± 0.06 [49]

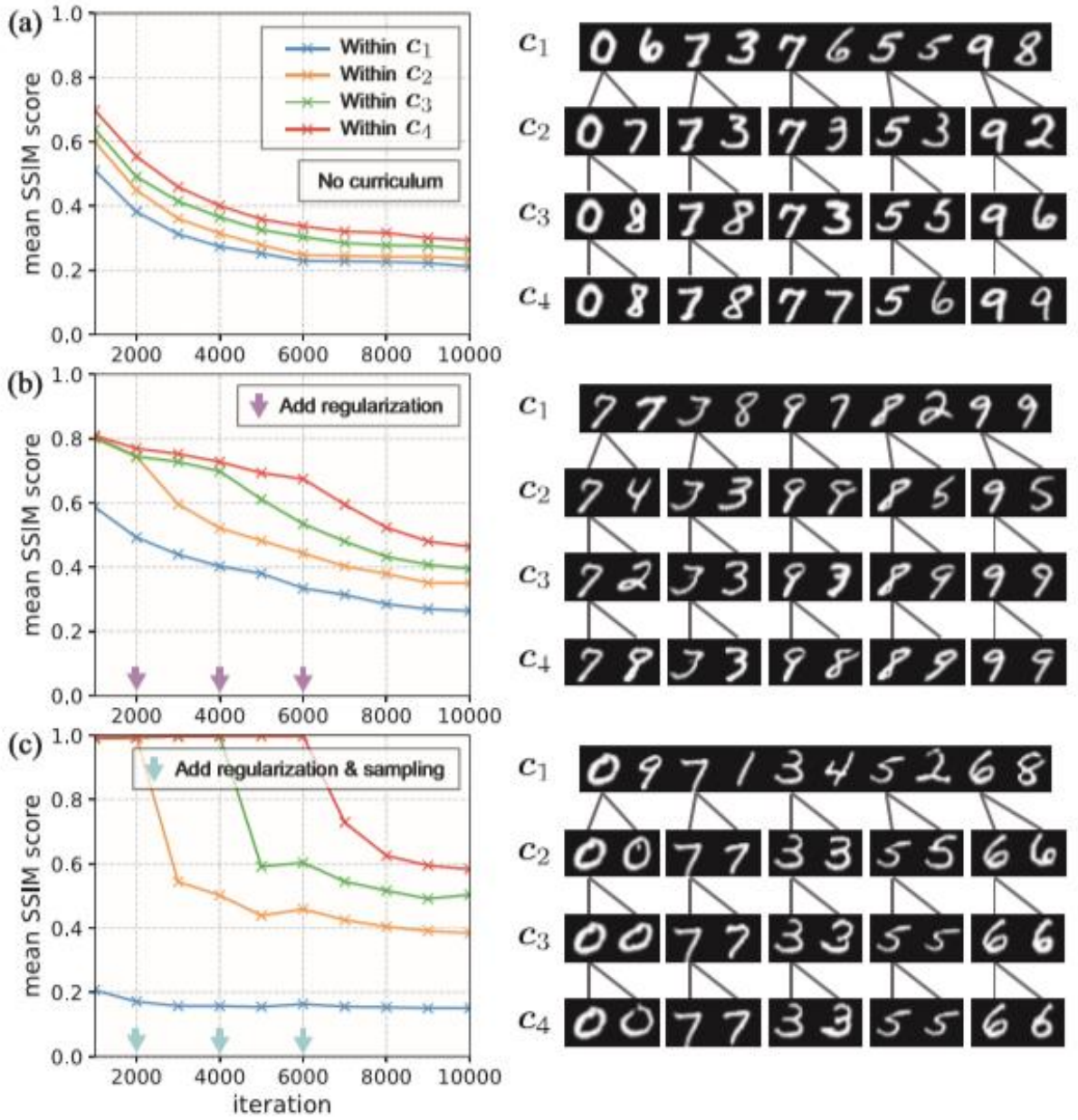
Table 2. Quantitative comparison between GAN, AC-GAN, and DTLC-GAN$_{WS}$

Model	CIFAR-10 (Unsupervised)	CIFAR-10 (Supervised)	Tiny ImageNet (Unsupervised)
WGAN-GP	$7.86 \pm .07$ [12]	-	$8.33 \pm .11$
AC/Info-WGAN-GP	$7.97 \pm .09$	$8.42 \pm .10$ [12]	$8.33 \pm .10$
DTLC^2-WGAN-GP	$8.03 \pm .12$	$8.44 \pm .10$	$8.34 \pm .08$
DTLC^3-WGAN-GP	$8.15 \pm .08$	$8.56 \pm .07$	$8.41 \pm .10$
DTLC^4-WGAN-GP	$\mathbf{8.22 \pm .11}$	$\mathbf{8.80 \pm .08}$	$\mathbf{8.51 \pm .08}$
State-of-the-Art	$7.86 \pm .07$ [12]	$8.59 \pm .12$ [13]	-

Table 3. Inception scores for WGAN-GP-based models

Ablation Study

SSIM: Smaller value larger diversity



Contributions

1. Enables semantic features of an image to be controlled in a coarse-to-fine manner.
2. We incorporate a new architecture called the DTLC into a GAN.
3. We propose a regularization called the HCMI to learn hierarchically disentangled representations only using a single DTLC-GAN model without relying on detailed supervision

HCMI: Hierarchical Conditional Mutual Information Regularization
DTLC: Decision Tree Latent Controller

Some thoughts/inspirations

Some thoughts – Information Gain

- Information gain usually used in *feature selection* task.
- Rare paper on Information Gain experiment with MNIST, CIFAR, SVHN

Some thoughts – CatGAN vs DTLC-GAN

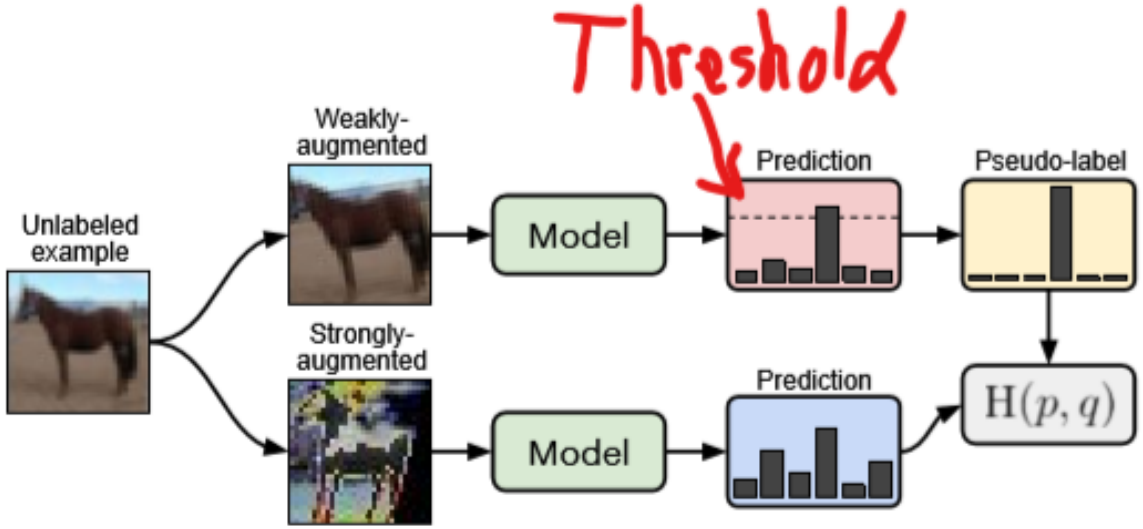
Usage of mutual information (MI):

- CatGAN: MI between input images and predicted label (Discriminator)
- DTLC-GAN/InfoGAN: MI between latent code and generated images (Generator)

Mission:

- CatGAN: Classification
- DTLC-GAN/InfoGAN: Generate fake images

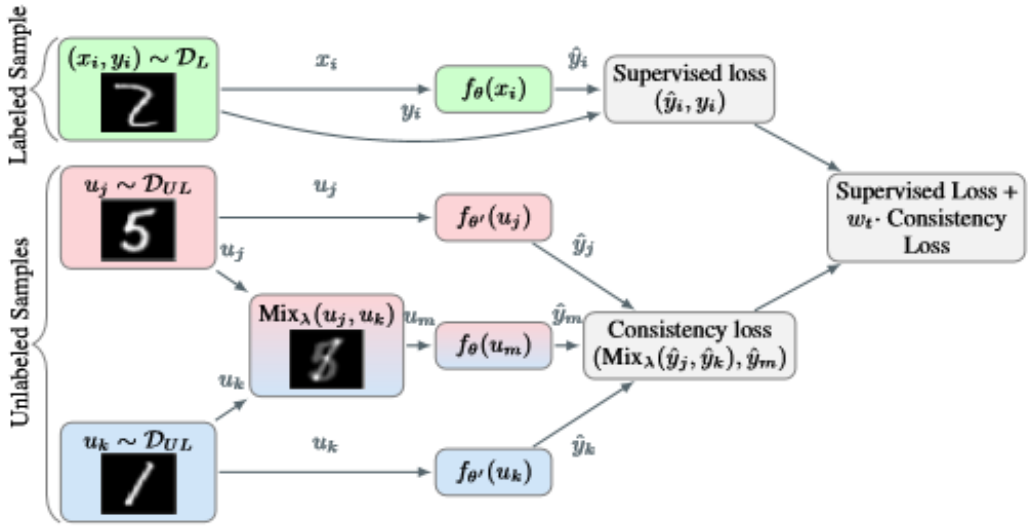
Inspirations – Fixmatch



Extract from paper Fixmatch

- Using GAN to generate fake images to have similar effect as in augmentation in Fixmatch.
- Consistency Regularization as proposed in Fixmatch. Used only *highly confident* fake images to calculate cross entropy. (Sample filtering)
- A *Bad GAN* is necessary.

Inspirations – Interpolation Consistency Training



Extract from paper Interpolation Consistency Training

- Most useful <u>consistency regularization</u> should applied on the samples <u>near *decision boundary*</u>(low density region).
- However, *random perturbations* is inefficient strategy.
- *Interpolation* is a good perturbation for consistency-based regularization

Some thoughts – GAN

- Previous work focus on generating images at low density region via feature matching method.
- Consistency regularization of localized GAN:
  - Implement with **noise**
  - Involve all samples instead of considering low density region only
- Should introduce **consistency regularization** on images at **low density** region <---- 可以尝试
- Consistency regularization on fake samples instead of true samples for