Semi-crowdsourced Clustering with Deep Generative Models

ZHUSUAN
github.com/thu-ml/zhusuan

Yucen Luo, Tian Tian, Jiaxin Shi, Jun Zhu, Bo Zhang
Department of Computer Science and Technology, Tsinghua University, Beijing, China
luoyc15@mails.tsinghua.edu.cn. **Code available at** `https://github.com/xinmei9322/semicrowd`

1. Learning from crowds

- Distribute micro-tasks to web workers in parallel, fast with relatively low cost
- Comparing pairs is easier for non-experts → pairwise constraints

2. (Semi-) crowd clustering

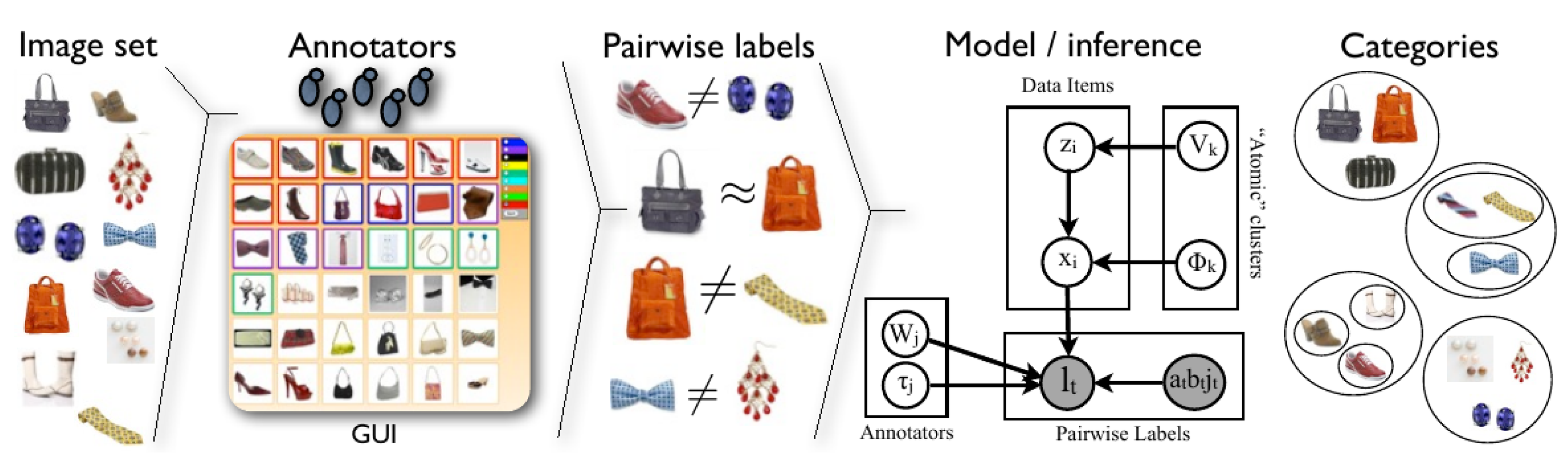
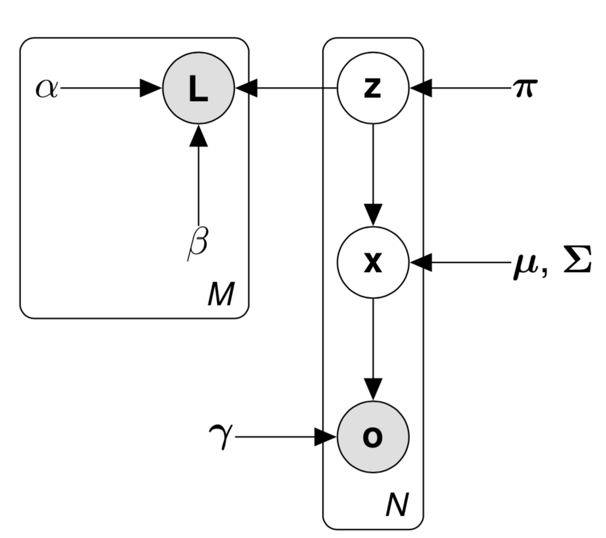


Figure 1: Schematic of Bayesian crowdclustering (from Gomes et al., NIPS 2011).

- **Bayesian crowdclustering** [Gomes et al., NIPS 2011]
  → Cost grows quadratically as N grows. Not scalable!
- **SemiCrowd** [Yi et al., NIPS 2012]
  → Linear similarity function, ignores the noise and inter-worker variations
- **Multiple Clustering Views from Multiple Uncertain Experts** [Chang et al., ICML 2017]
  → Discriminative clustering, does not use the information in unlabeled samples

4. Simple version: Amortized inference



- Let $\Theta = \{\pi, \mu, \Sigma, \alpha, \beta\}$, maximize the variational lower bound $\mathcal{L}$

$$\log p(\mathbf{O}, \mathbf{L}) \geq \mathbb{E}_{q(\mathbf{Z},\mathbf{X}|\mathbf{O})}[\log p(\mathbf{Z}, \mathbf{X}, \mathbf{O}, \mathbf{L}; \Theta, \gamma) - \log q(\mathbf{Z}, \mathbf{X}|\mathbf{O})] = \mathcal{L}(\mathbf{O}, \mathbf{L}; \Theta, \gamma, \phi)$$

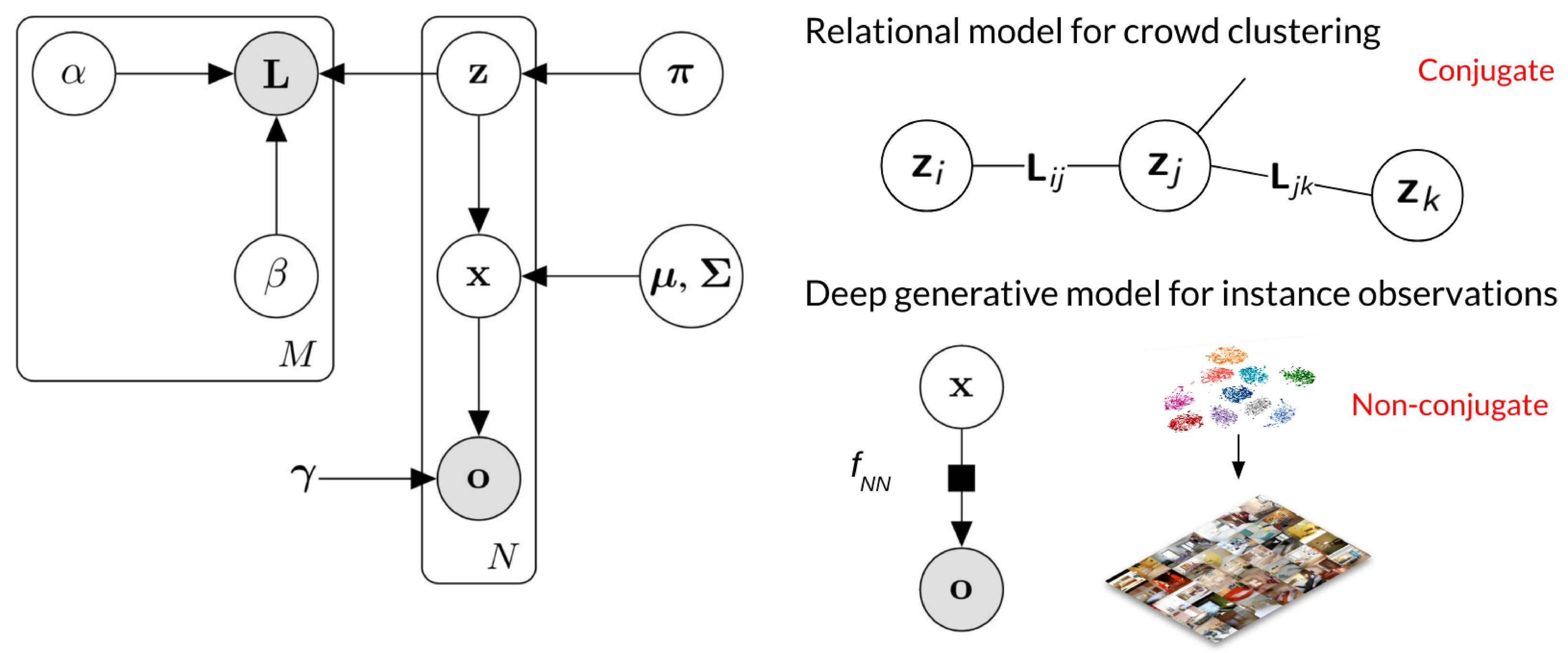
- Inference networks

$$q(\mathbf{z}_n|\mathbf{o}_n; \phi) = \mathrm{Cat}(\mathbf{z}_n; \boldsymbol{\pi}(\mathbf{o}_n; \phi)),$$

$$q(\mathbf{x}_n|\mathbf{z}_n, \mathbf{o}_n; \phi) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{z}_n, \mathbf{o}_n; \phi), \mathrm{diag}(\boldsymbol{\sigma}^2(\mathbf{z}_n, \mathbf{o}_n; \phi))),$$

- Analytically sum over the discrete $\mathbf{z}_n$; use the reparameterization trick for $\mathbf{x}_n$.

3. Guide the learning of DGMs with statistical relational models



- **DGM:** Raw data observations $\mathbf{o}_n$, corresponding latent variable $\mathbf{x}_n$, cluster index $\mathbf{z}_n$.

$$p(\mathbf{Z}; \boldsymbol{\pi}) = \prod_{n=1}^{N}\prod_{k=1}^{K} \pi_k^{z_{nk}}, \; p(\mathbf{X}|\mathbf{Z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N}\prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}},$$

$$p(\mathbf{O}|\mathbf{X}; \gamma) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{o}_n|\boldsymbol{\mu}_\gamma(\mathbf{x}_n), \mathrm{diag}(\boldsymbol{\sigma}^2_\gamma(\mathbf{x}_n))),$$

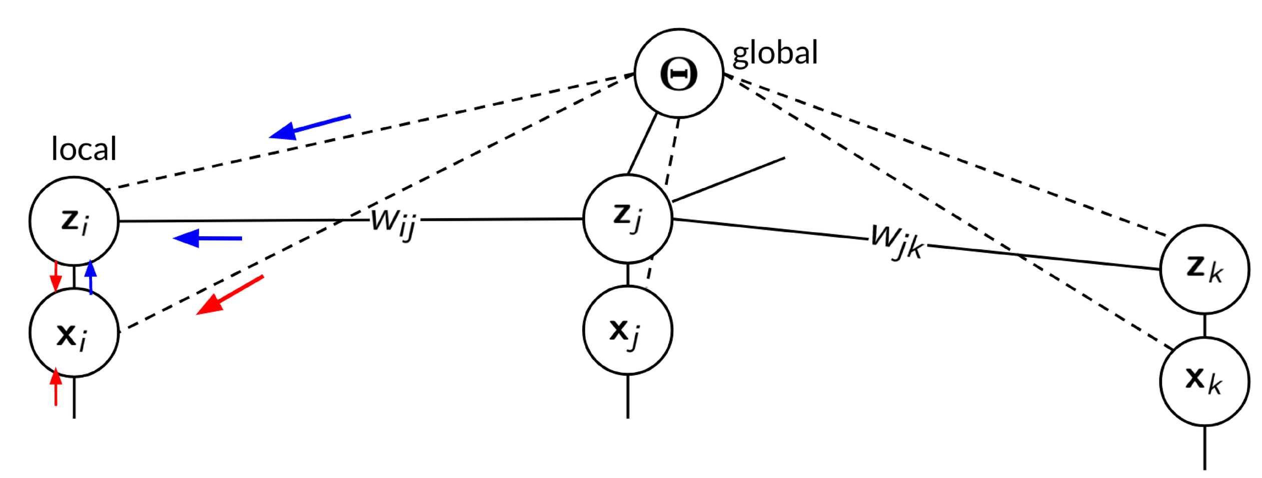
- **Relational model:** M workers, accuracy parameters: sensitivity $\boldsymbol{\alpha}$ and specificity $\boldsymbol{\beta}$

$$p(L_{ij}^{(m)}|\mathbf{z}_i, \mathbf{z}_j; \alpha_m, \beta_m) = \mathrm{Bern}(L_{ij}^{(m)}|\alpha_m)^{\mathbf{z}_i^\top \mathbf{z}_j} \mathrm{Bern}(L_{ij}^{(m)}|1-\beta_m)^{1-\mathbf{z}_i^\top \mathbf{z}_j},$$

5. Natural gradient stochastic variational inference

- Variational message passing for conjugate structures and amortized learning of deep components
- **Advantages**: (1) automatically determine model complexity (2) no need to sum over **z**
- Replacing the non-conjugate $p(\mathbf{O}|\mathbf{X}; \gamma)$ using recognition networks $r(\mathbf{o}_i; \phi)$.

$$\widehat{\mathcal{L}}(\eta_\Theta, \eta_Z, \eta_X; \phi) \triangleq \mathbb{E}_{q(\Theta,\mathbf{Z},\mathbf{X})} \log\left[\frac{p(\mathbf{L}^{(1:M)}, \mathbf{X}, \mathbf{Z}, \Theta)\exp\{\langle r(\mathbf{o}_i; \phi), \mathbf{t}(\mathbf{x}_i)\rangle\}}{q(\Theta)q(\mathbf{Z})q(\mathbf{X})}\right].$$



- Local partial optimizers for $\widehat{\mathcal{L}}$: $q^*(\mathbf{X}) = \prod_{i=1}^N q^*(\mathbf{x}_i)$, $q^*(\mathbf{Z}) = \prod_{i=1}^N q^*(\mathbf{z}_i)$

$$\boldsymbol{\eta}^*_{\mathbf{x}_i} = \mathbb{E}_{q(\boldsymbol{\mu},\boldsymbol{\Sigma})}[\boldsymbol{\eta}^0_{\mathbf{x}_i}(\boldsymbol{\mu}, \boldsymbol{\Sigma})]^\top \mathbb{E}_{q(\mathbf{z}_i)}[\mathbf{t}(\mathbf{z}_i)] + r(\mathbf{o}_i; \phi),$$

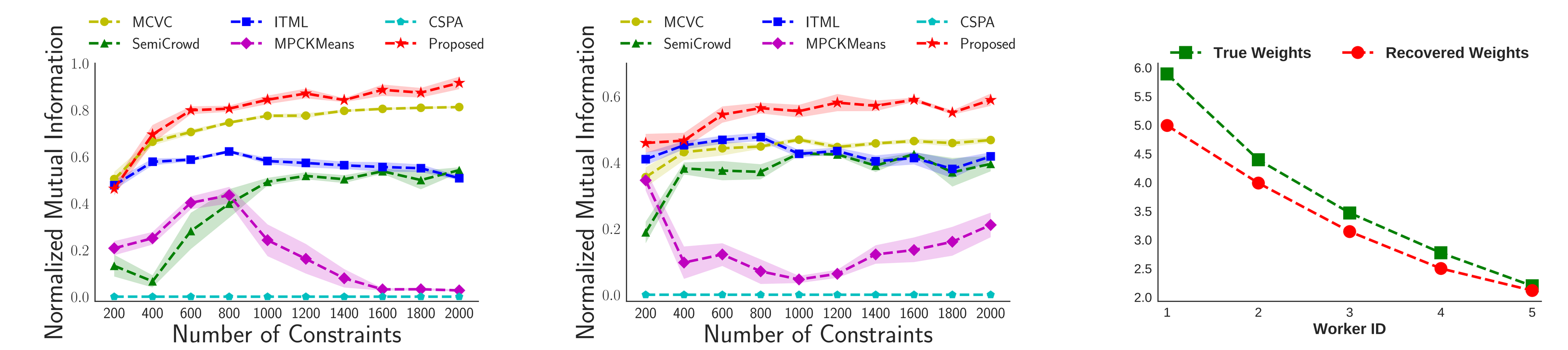
$$\boldsymbol{\eta}^*_{\mathbf{z}_i} = \mathbb{E}_{q(\boldsymbol{\pi})}\mathbf{t}(\boldsymbol{\pi}) + \mathbb{E}_{q(\boldsymbol{\mu},\boldsymbol{\Sigma})}[\mathbf{t}(\boldsymbol{\mu}, \boldsymbol{\Sigma})]^\top \mathbb{E}_{q(\mathbf{x}_i)}[(\mathbf{t}(\mathbf{x}_i), 1)] + \sum_{m=1}^{M}\sum_{j=1}^{N} w_{ij}^{(m)} \mathbb{E}_{q(\mathbf{z}_j)}[\mathbf{t}(\mathbf{z}_j)],$$

where $w_{ij}^{(m)} = I_{ij}^{(m)} \mathbb{E}_{q(\boldsymbol{\alpha},\boldsymbol{\beta})}\left[\ln\frac{1-\alpha_m}{\beta_m} + L_{ij}^{(m)}\left(\ln\frac{\alpha_m}{1-\alpha_m} + \ln\frac{\beta_m}{1-\beta_m}\right)\right]$.

- Final objective: $\mathcal{J}(\boldsymbol{\eta}_\Theta; \phi, \gamma) \triangleq \mathcal{L}(\boldsymbol{\eta}_\Theta, \boldsymbol{\eta}^*_Z(\boldsymbol{\eta}_\Theta, \phi), \boldsymbol{\eta}^*_X(\boldsymbol{\eta}_\Theta, \phi), \gamma)$.
- Update the global variational parameters $\boldsymbol{\eta}_\Theta$ by natural gradients: $\widetilde{\nabla}_{\eta_\Theta}\mathcal{J}$
- For other parameters ϕ, γ, compute the gradients $\nabla_\phi \mathcal{J}(\boldsymbol{\eta}_\Theta; \gamma, \phi)$ and $\nabla_\gamma \mathcal{J}(\boldsymbol{\eta}_\Theta; \gamma, \phi)$.

6. Outperforms competing methods

- Face dataset, 640 images from 20 people with different poses (straight, left, right, up).



(a) All the data points are annotated (b) Only 100 data points are annotated (c) True accuracies are set to $\alpha = \beta = [0.95, 0.9, 0.85, 0.8, 0.75]$.

7. Crowdsourced real annotations from Amazon Mechanical Turks

Method	without annotations			with annotations		
	Accuracy	NMI	Time	Accuracy	NMI	Time
SCDC	65.92 ± 3.47 %	0.6953 ± 0.0167	177.3s	81.87 ± 3.86%	0.7657 ± 0.0233	201.7s
BayesSCDC	77.64 ± 3.97 %	0.7944 ± 0.0178	11.2s	**84.24 ± 5.52%**	**0.8120 ± 0.0210**	16.4s

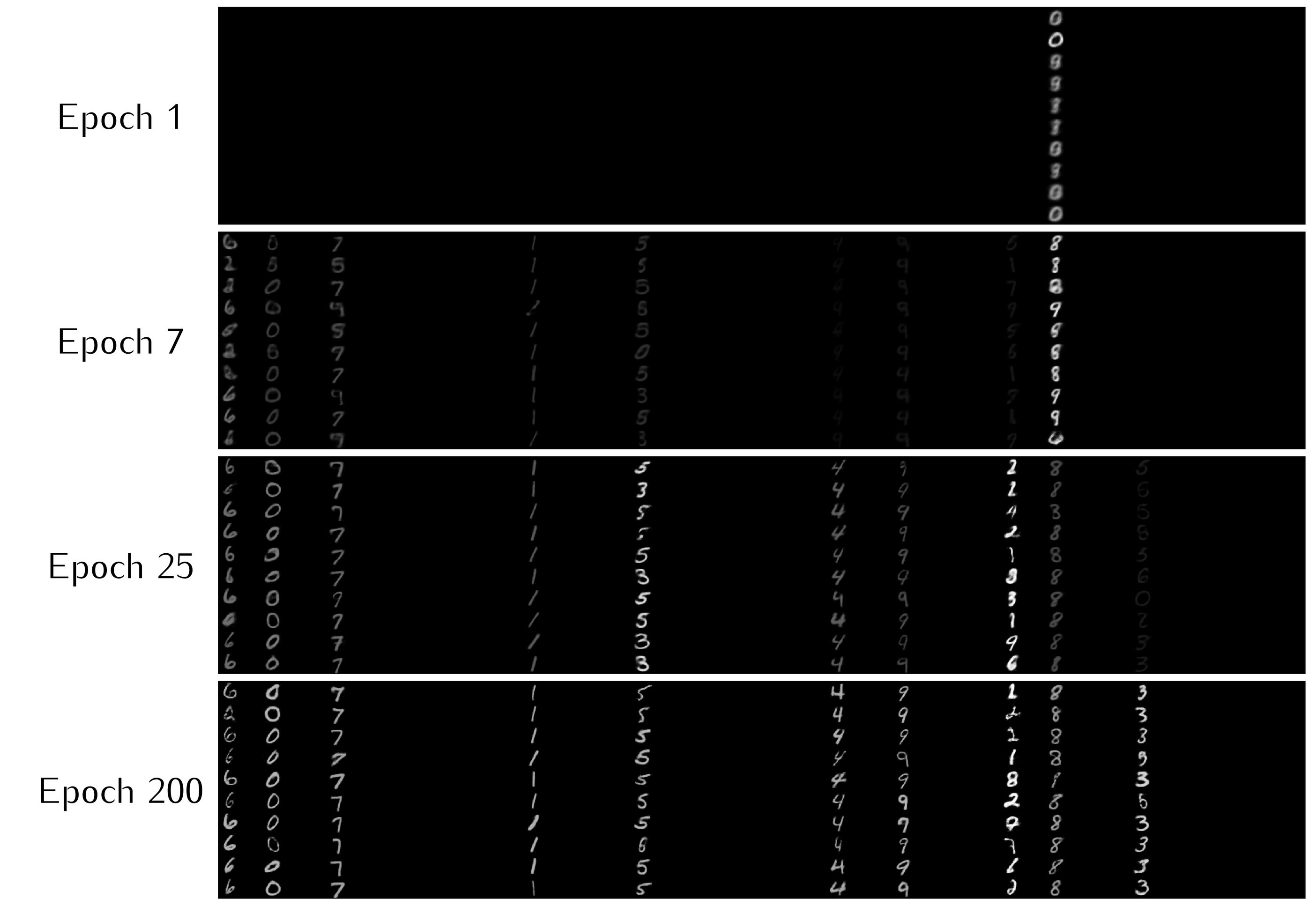


Figure 3: MNIST: visualization of generated samples of 50 clusters during training BayesSCDC. Each column represents a cluster, whose inferred proportion (π_k) is reflected by brightness.

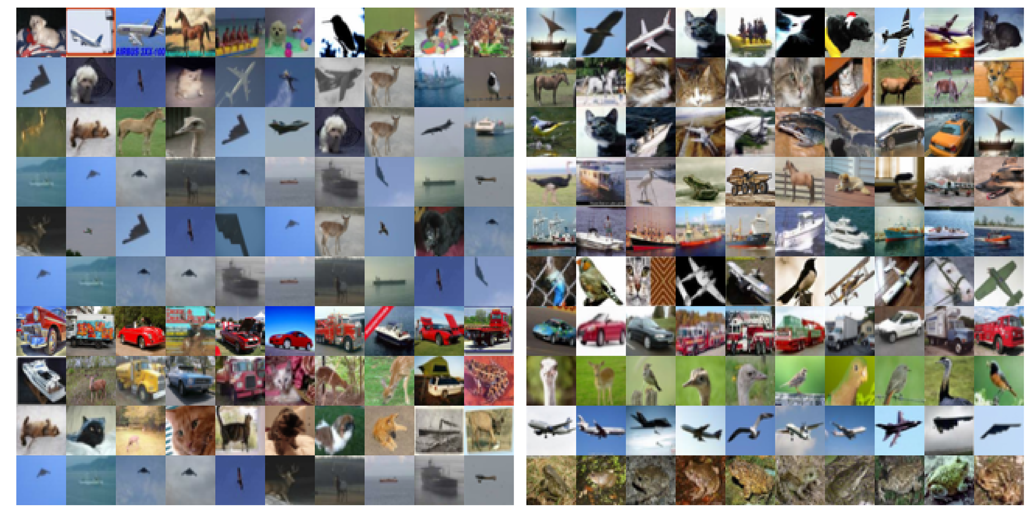


Figure 4: Clustering results on CIFAR-10: (left) unsupervised; (right) with noisy annotations.