Knowledge Extraction with No Observable Data

Jaemin Yoo
jaeminyoo@snu.ac.kr

Minyong Cho
chominyong@gmail.com

Taebum Kim
k.taebum@snu.ac.kr

U Kang
ukang@snu.ac.kr

Data Mining Laboratory

Seoul National University

Summary

KEGNET (Knowledge Extraction with Generative Networks)

- **Input:** a trained neural network M without data
- **Output:** a generator G that estimates unknown p_x
- **Main idea:** G is trained as a function $(y, z) \rightarrow x$
- **GitHub:** https://github.com/snudatalab/KegNet

Knowledge Extraction

Research motivation

- *A trained network is given, but no data available*
- *How can we distill the knowledge without data?*
- It is intractable to estimate directly $p_x(x)$
- Estimate $p(x|y,z)$ given random variables y and z
  - y is a probability vector representing a label
  - z is a low-dimensional embedding vector of data

Objective function

- Generate artificial data examples:

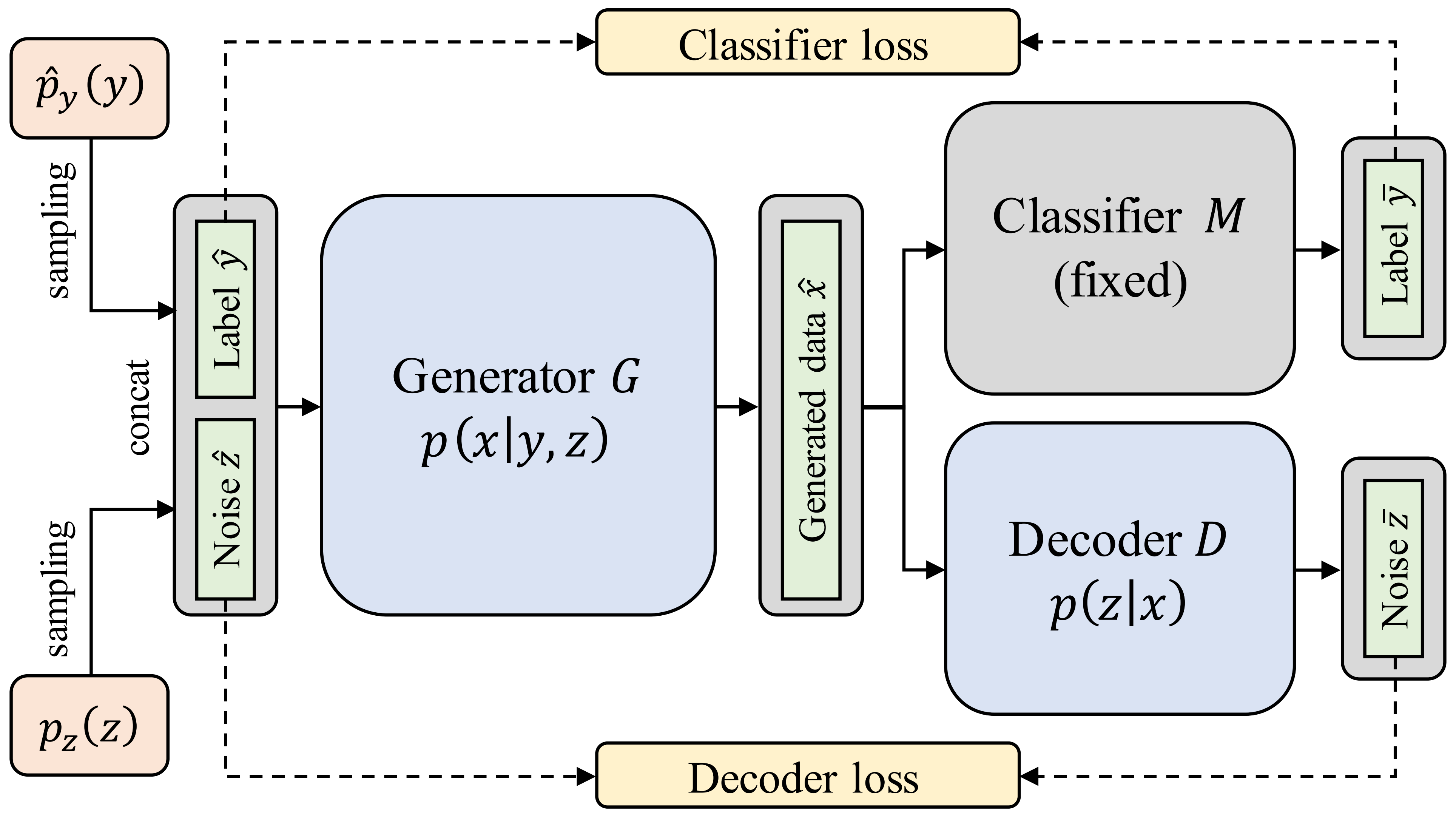
$$\mathcal{D} = \left\{ \underset{\hat{x}}{\operatorname{argmax}}\, p(\hat{x}|\hat{y},\hat{z}) \;\middle|\; \hat{y} \sim \hat{p}_y(y) \text{ and } \hat{z} \sim p_z(z) \right\}$$

- The argmax function is approximated as follows:

$$\underset{\hat{x}}{\operatorname{argmax}}\, p(\hat{x}|\hat{y},\hat{z}) \approx \underset{\hat{x}}{\operatorname{argmax}}(\log p(\hat{y}|\hat{x}) + \log p(\hat{z}|\hat{x}))$$

- Thus, we have two reconstruction terms for $\hat{y}$ and $\hat{z}$

Proposed Architecture



Classifier M

- Given and fixed; our only evidence for estimation
- *LeNet4* or *ResNet14* in our experiments

Generator G

- Estimate $p(x|y,z)$ by a generator network
- Its structure is based on *ACGAN* in our experiments
- Classifier loss makes $M(G(\hat{y}))$ similar to $\hat{y}$

Decoder D

- Estimate $p(z|x)$ to find the meaning of $\hat{x}$
- Increase the variance of various $\hat{x}$ given the same $\hat{y}$
- Decoder loss makes $D(G(\hat{y}))$ similar to $\hat{z}$

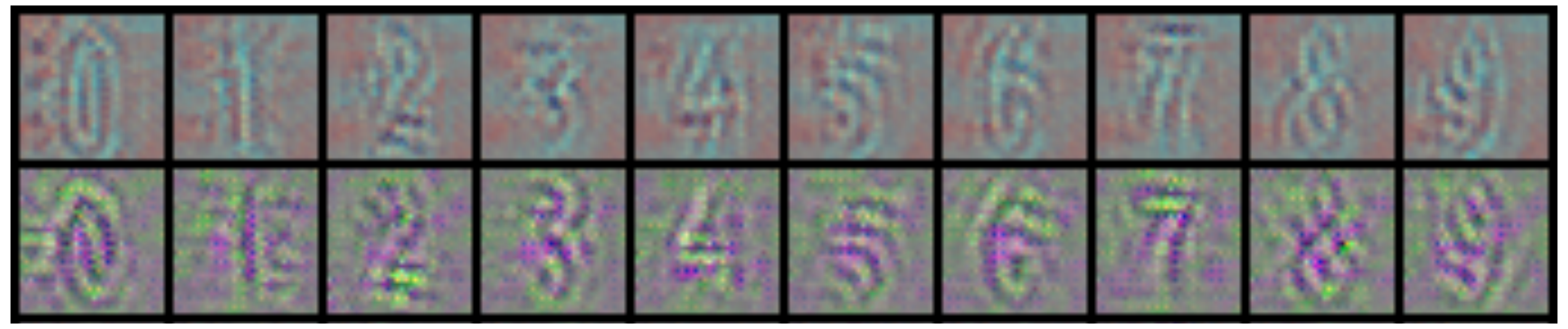
Experiments

Data-free model compression

- **Models:** *LeNet5* and *ResNet14*
- **Datasets:** *MNIST*, *SVHN*, and *Fashion MNIST*
- **Competitors**
  - Tucker (T): Tucker decomposition without fine-tuning
  - T+Uniform: Estimate p_x as the uniform dist. $\mathcal{U}(-1, 1)$
  - T+Gaussian: Estimate p_x as the normal dist. $\mathcal{N}(0, 1)$

Dataset	Model	Approach	Student 1	Student 2
SVHN	ResNet14	Original	93.23%	93.23%
SVHN	ResNet14	Tucker (T)	19.31% (1.44×)	11.02% (1.65×)
SVHN	ResNet14	T+Uniform	33.08 ± 1.47%	63.08 ± 1.77%
SVHN	ResNet14	T+Gaussian	26.58 ± 1.61%	60.22 ± 4.17%
SVHN	ResNet14	T+**KEGNET**	**69.89 ± 1.24%**	**87.26 ± 0.46%**

Generated images for SVHN from two generators



Latent space walking from label 0 to label 5 in SVHN