Label-Noise Robust Twin Auxiliary Classifier GANs

Michael Law

University of Melbourne

June 30, 2020

Table of Contents

Generative Networks
- Generative Adversarial Networks (GAN)
- Auxiliary Classifier GAN (AC-GAN)
- Twin Auxiliary Classifier GAN (TAC-GAN)

Generative Networks with Label Noise
- Existing Methods: rAC-GAN and RC-GAN
- Classification with Label Noise: Co-Teaching
- Proposed Method: CO-TAC
- Classification with Label Noise: MentorNet
- Proposed Method: TAC-MENTOR

Experiments

Conclusion and Future Work

Generative Networks - Overview

- Using artificial neural networks to **generate** data, as opposed to classifying data
- Typical scenario: low-dimensional vector is sampled from a **latent space**, e.g. a multivariate normal distribution
- A generative network maps this to a high-dimensional output (e.g. images)

Generative Networks - Applications

- Image generation

Source: Brock et al., Large Scale GAN Training for High Fidelity Natural Image Synthesis

Generative Networks - Applications

- Scenery generation from segmentation maps

Source: Wang et al., High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs

Generative Networks - Applications

- Chatbots

User: `Does money buy happiness?`
Bot: `Depends how much money you spend on it.`
User: `What is the best way to buy happiness?`
Bot: `You just have to be a millionaire by your early 20s, then you can be happy.`
User: `This is so difficult!`
Bot: `You have no idea how hard it is to be a millionaire and happy. There is a reason the rich have a lot of money.`

Source: Zhang et al., DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation

Generative Networks - Applications

- Generating images from text prompts

this small bird has a pink breast and crown, and black primaries and secondaries.

this magnificent fellow is almost all black with a red crest, and white cheek patch.

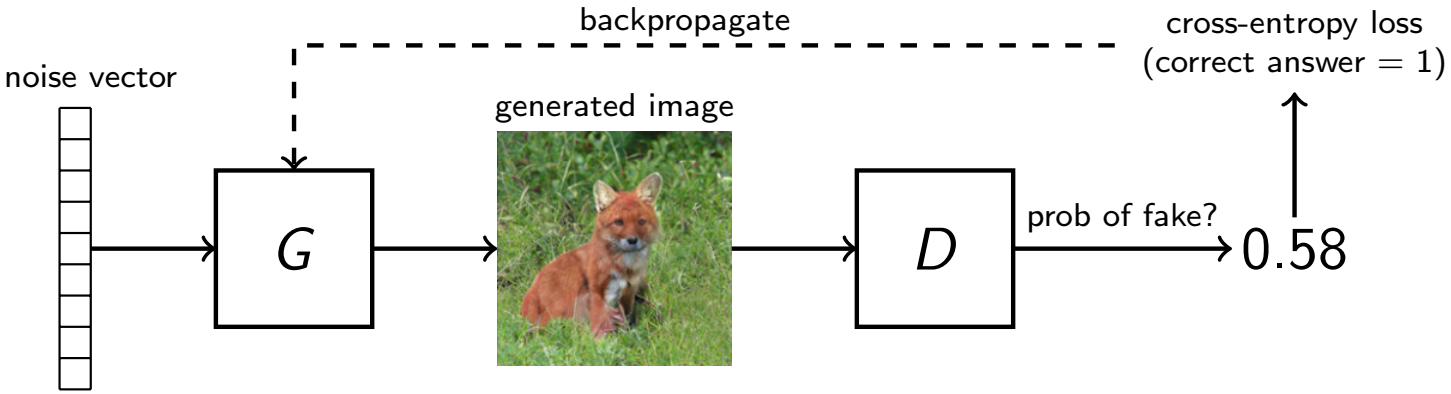
Source: Reed et al., Generative Adversarial Text to Image Synthesis

Generative Adversarial Networks (GAN)

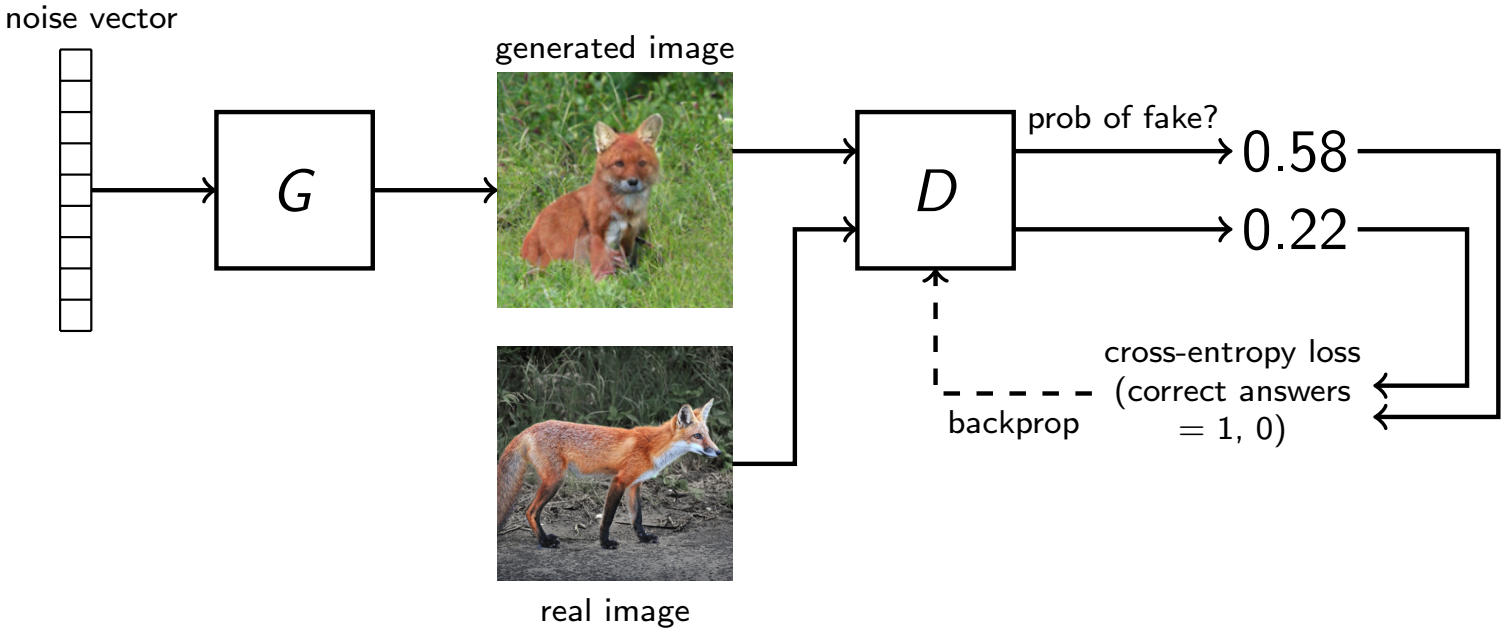
- A framework for generative networks, popular for image generation
- In its most basic form, GAN consists of two networks:
  - **Generator** G**:** deconvolutional network mapping random noise vectors to high-dimensional images
  - **Discriminator** D**:** convolutional network deciding whether images are real or fake (generated by G)

GAN - training algorithm

- Generator (G) training step

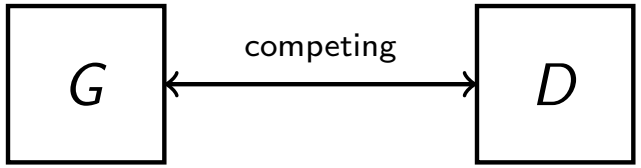


- Discriminator (D) training step



GAN

- Networks improve through **competition**
  - D learns to distinguish between real and fake images
  - G generates better images to fool D, attempting to incapacitate D from distinguishing between real/fake



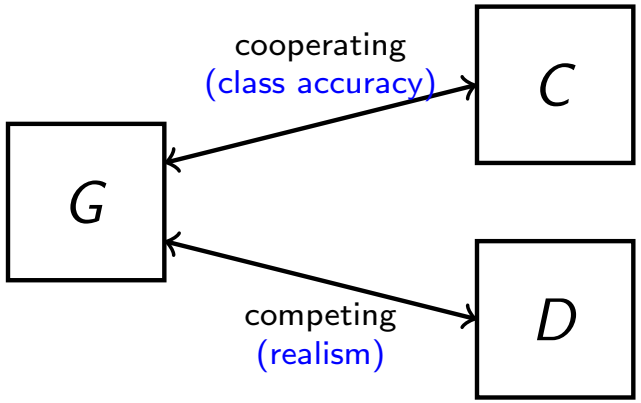
- The GAN objective function reflects this competition:

$$\min_G \max_D \mathcal{L}(D, G) = \mathbb{E}_{\vec{x}\sim p_X}[\ln D(\vec{x})] + \mathbb{E}_{\vec{z}\sim N(0,I)}[\ln(1 - D(G(\vec{z}))].$$

- To endow GANs with additional capabilities, add more networks into the GAN 'ecosystem'

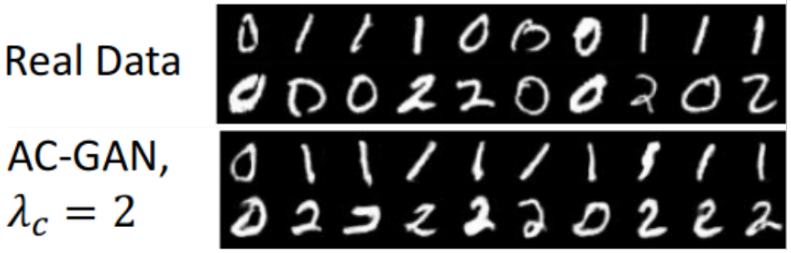
Auxiliary Classifier GAN (AC-GAN)

- Naturally, one would be interested in generating specific types of images, e.g. of certain object classes
- AC-GAN includes an auxiliary classifier to C enable class-conditional image generation
- Training data consists of (image, label) pairs



Twin Auxiliary Classifier GAN (TAC-GAN)

- AC-GAN is vulnerable to mode collapse when there exist overlaps between image classes ('class support overlaps')
  - Overlapping MNIST

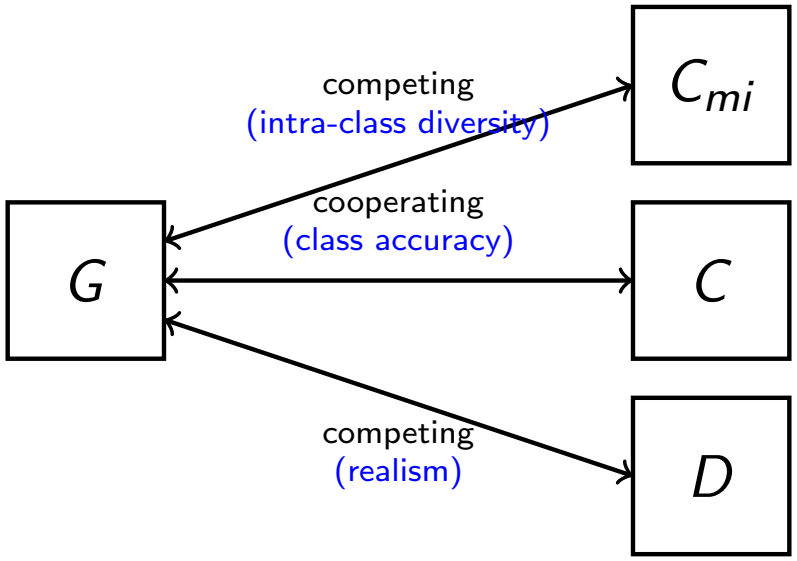


  - VGGFace2

Source: Gong, Xu et al., Twin Auxiliary Classifiers GAN

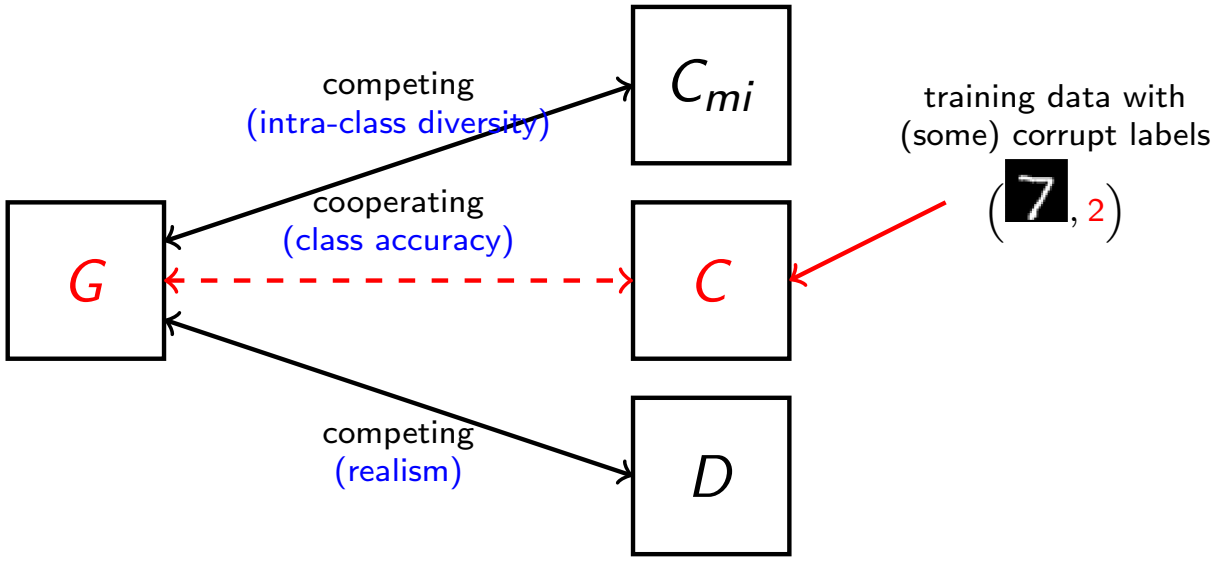
Twin Auxiliary Classifier GAN (TAC-GAN)

- Solution: Twin Auxiliary Classifier GAN
- TAC-GAN adds another auxiliary classifier C_{mi} which *competes* with G, encouraging G to generate images with greater diversity *within* classes



Generative Networks with Label Noise

- What if (image, label) pairs in the training set contain *incorrect* labels?
- None of the GAN variants we have mentioned are robust to label noise. E.g. for TAC-GAN:



Existing Methods: rAC-GAN and RC-GAN

- Assumes the existence of a noise transition matrix T, $T_{ij} =$ probability of a class-i image being (possibly incorrectly) labelled as class j
- These methods assume knowledge of, or estimate, the transition matrix
- Idea: use $\hat{T}$ to corrupt labels supplied to G, or corrupt the outputs of the auxiliary classifier C. Use a modified loss function to encourage G to generate images of correct classes.

Existing Methods: rAC-GAN and RC-GAN

These approaches are somewhat successful, but some concerns arise:

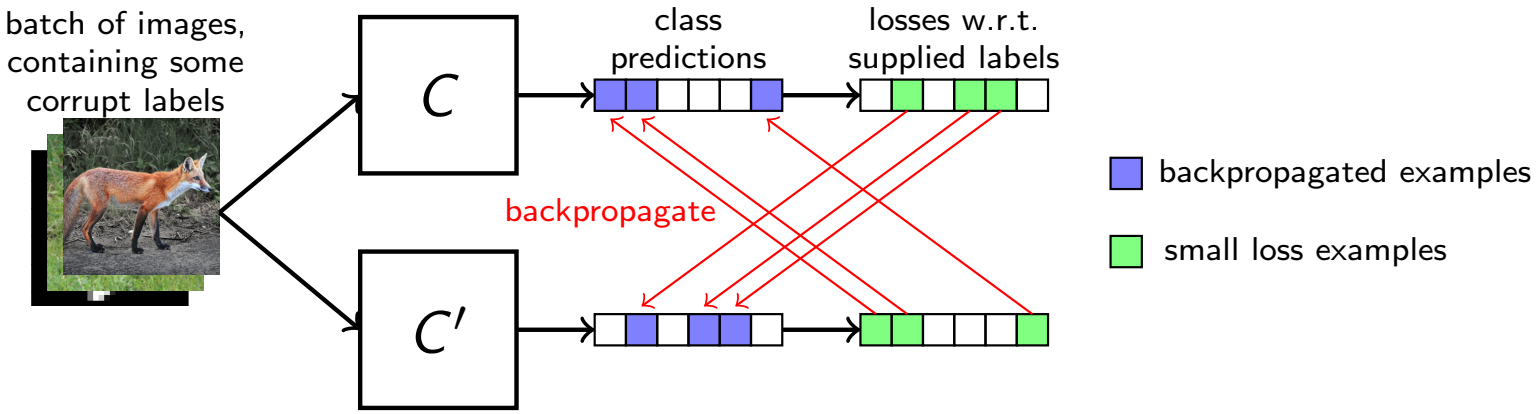
- In the real world, is it reasonable to expect one to know T?
- If T is to be estimated, is it easy to estimate?
  - If the number of classes is n, then T contains n^2 entries
- Does such a T exist in the first place?
  - Label corruption probabilities vary from image to image within the same ground-truth class

These questions prompt us to develop matrix-free approaches to image generation with label noise.

How about we explore what solutions currently exist for *classification* with label noise, and 'port' these into TAC-GAN?

Classification with Label Noise: Co-Teaching

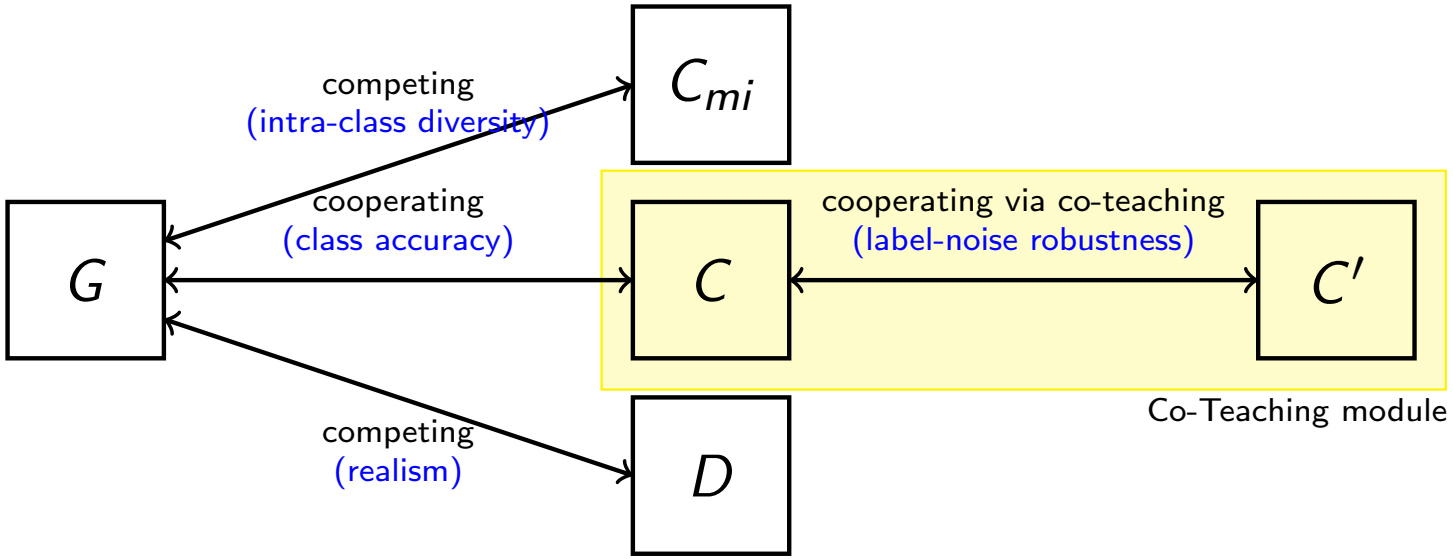
- Two classifiers *teach each other*. Each selects training examples it has achieved small losses on, and instructs the other to backpropagate only on those examples



- Gradually decrease number of chosen examples per minibatch as training progresses

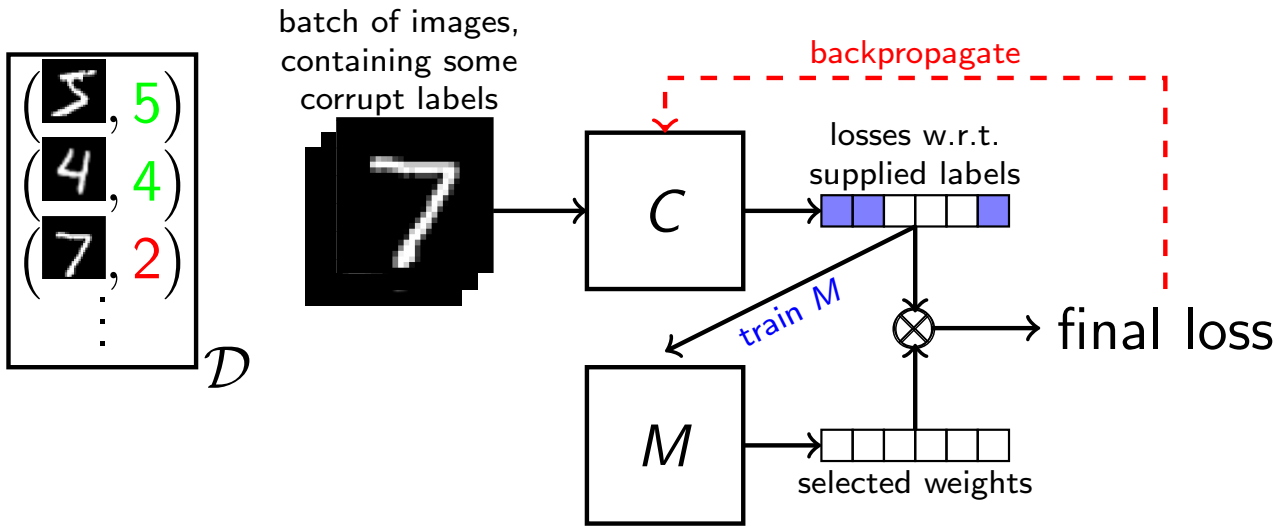
Generation with Label Noise: Proposed Method 1

- TAC-GAN with Co-Teaching (CO-TAC)



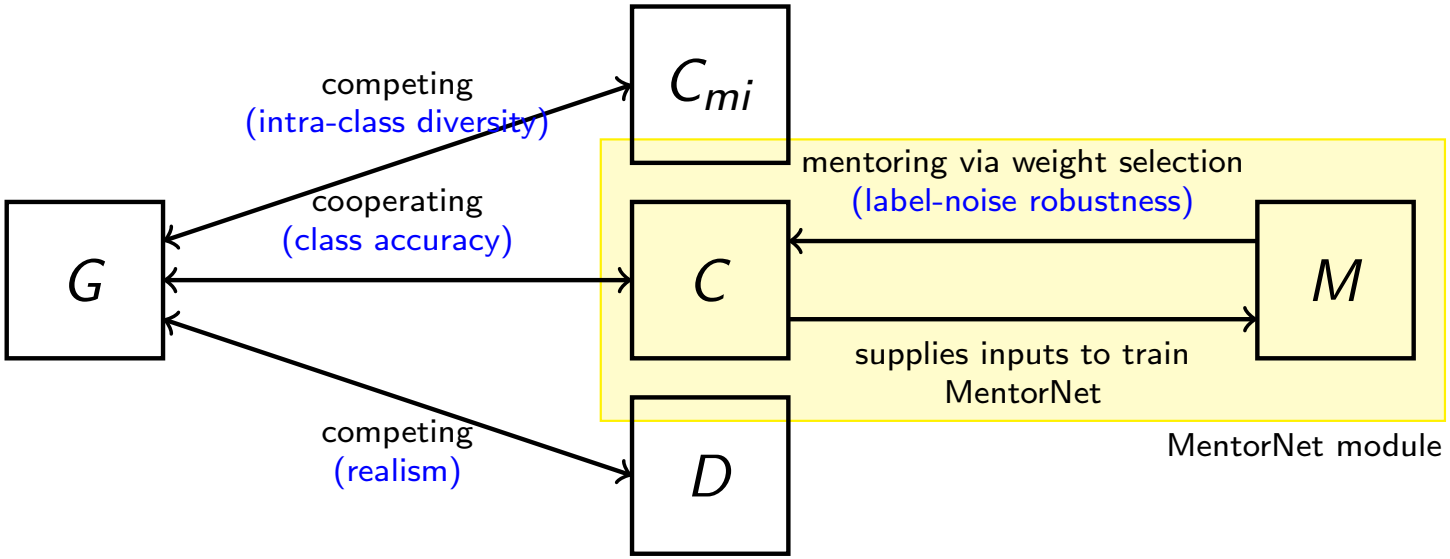
Classification with Label Noise: MentorNet

- A network M (*MentorNet*) selects C's backpropagation weights for each training example, effectively adjusting each example's 'learning importance'
- Assumes knowledge of label (in)correctness for a small portion $\mathcal{D}$ of the whole dataset (reasonable?)
- M is trained using C's input features e.g., loss, loss percentile in minibatch, for examples in $\mathcal{D}$



Generation with Label Noise: Proposed Method 2

- TAC-GAN with MentorNet (TAC-MENTOR)



Methods to Experiment

- Baseline RC-TAC-G (RC-GAN adapted to TAC-GAN)
- Baseline: RC-TAC-C (rAC-GAN adapted to TAC-GAN)
- Proposed: CO-TAC (TAC-GAN with Co-Teaching)
- Proposed: TAC-MENTOR (TAC-GAN with MentorNet)

The proposed methods are transition matrix-free, but they can be programmed to assume *nothing* about the label noise structure!

Experiments: MNIST

- Uniform corruption with probability c: probability $c/(n-1)$ of corrupting to each of the wrong class labels
- Transition matrix: $\begin{bmatrix} 1-c & \frac{c}{n-1} & \cdots & \frac{c}{n-1} \\ \frac{c}{n-1} & 1-c & \cdots & \frac{c}{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{c}{n-1} & \frac{c}{n-1} & \cdots & 1-c \end{bmatrix}$
- GAN-test: accuracy of pretrained classifier on generated images. GAN-train: accuracy of C_{mi} on real images.

MNIST	**Uniform**									
Corruption prob c	0.20		0.35		0.50		0.65		0.80	
GAN-test or -train (%)	test	train	test	train	test	train	test	train	test	train
RC-TAC-G	82.9	98.9	78.7	98.8	63.8	98.8	51.0	98.2	36.4	96.5
RC-TAC-C	89.9	97.6	81.1	98.8	73.2	97.4	89.3	97.8	68.5	95.9
CO-TAC	97.5	97.2	96.0	97.9	94.4	98.2	98.1	92.8	72.2	78.9
TAC-MENTOR	97.4	99.1	93.8	97.9	96.4	98.6	94.7	97.9	93.5	95.2

Table 1: MNIST uniform corruption GAN-test and GAN-train results. Higher score is better.

Experiments: MNIST

- Flip corruption with probability c: probability c of corrupting to a certain class label ($1 \leftrightarrow 7$, $2 \leftrightarrow 5$, $3 \leftrightarrow 8$, $4 \leftrightarrow 9$, $6 \leftrightarrow 0$)
- Transition matrix: for the first row, $T_{00} = 1 - c$, $T_{06} = c$, else 0, etc.

MNIST	**Flip**							
Corruption prob c	0.15		0.25		0.35		0.45	
GAN-test or -train (%)	test	train	test	train	test	train	test	train
RC-TAC-G	82.2	96.1	78.4	98.1	63.2	97.1	54.2	72.0
RC-TAC-C	89.3	98.3	73.5	70.1	79.1	78.9	47.5	52.5
CO-TAC	98.1	98.2	95.0	96.8	94.1	95.2	93.6	97.2
TAC-MENTOR	95.9	98.9	93.4	98.4	81.2	97.9	61.4	84.9

Table 2: MNIST label-flip corruption GAN-test and GAN-train results. Higher score is better.

Experiments: MNIST

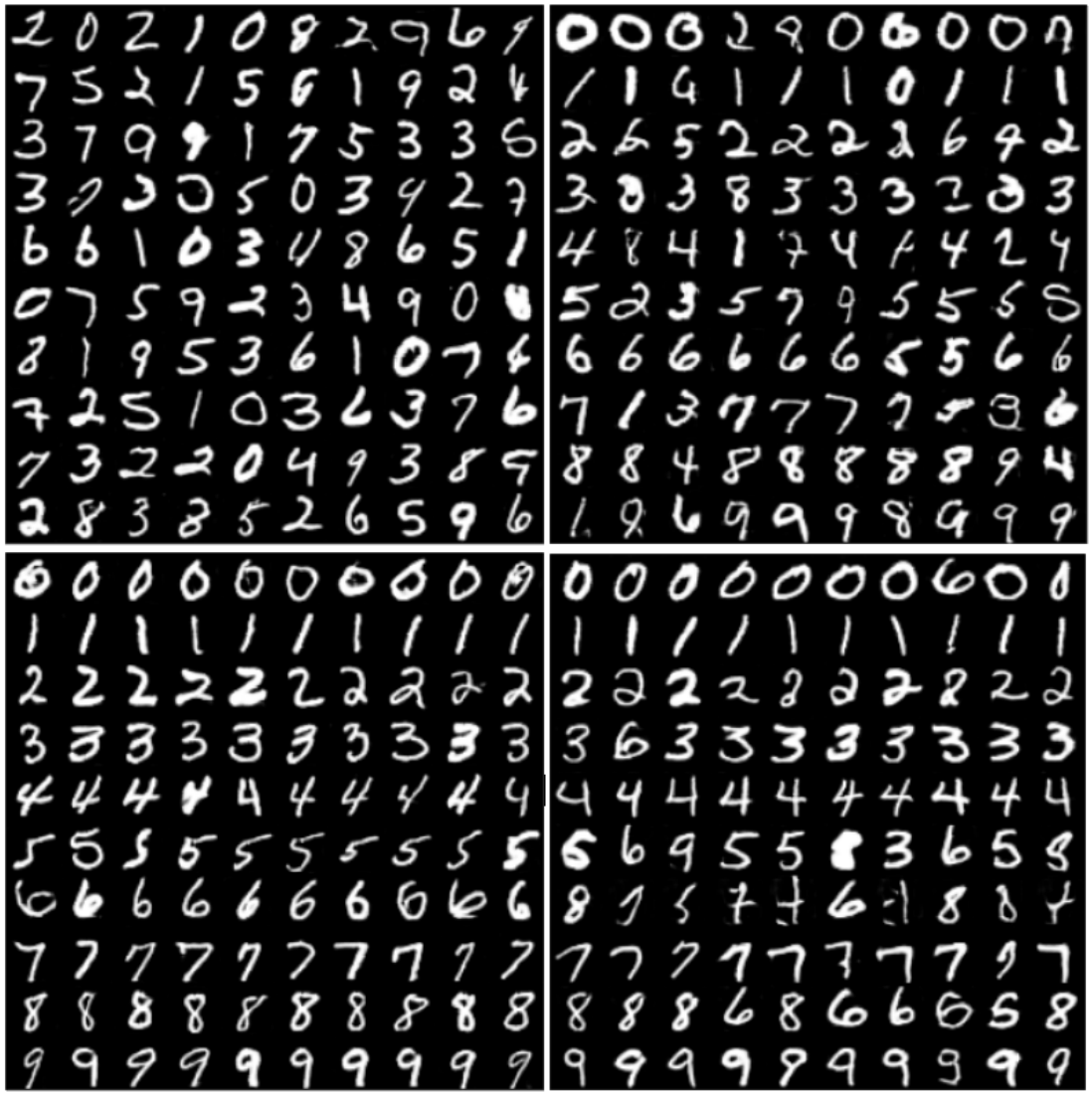


Figure 1: Generated image previews for MNIST with uniform corruption, where label corruption probability of training data = 0.8. Clockwise from top left: `RC-TAC-G`, `RC-TAC-C`, `CO-TAC`, `TAC-MENTOR`.

Experiments: CIFAR10

- FID: measures the similarity between a real image set and a generated image set. The lower, the better.
- Intra-FID: average FID over all classes

CIFAR10	**Uniform**									
Corruption prob c	0.20		0.35		0.50		0.65		0.80	
Intra-FID/GAN-train	I-FID	train	I-FID	train	I-FID	train	I-FID	train	I-FID	train
RC-TAC-G	19.8	82.5	36.3	80.1	73.0	69.7	71.7	58.2	93.7	64.3
RC-TAC-C	27.7	80.5	28.4	81.3	55.5	69.0	86.3	60.4	99.9	61.1
CO-TAC	24.4	81.9	29.4	82.2	39.1	80.6	59.3	79.0	66.8	73.1
TAC-MENTOR	20.6	81.3	30.1	78.5	54.5	75.6	97.9	71.6	96.2	65.3

Table 3: CIFAR10 uniform corruption Intra-FID and GAN-train results. Intra-FID: lower is better. GAN-train: higher is better.

Experiments: CIFAR10

- Label flips used: (airplane, bird), (automobile, truck), (cat, dog), (deer, horse), (frog, ship).

CIFAR10	**Flip**							
Corruption prob c	0.15		0.25		0.35		0.45	
Intra-FID/GAN-train	I-FID	train	I-FID	train	I-FID	train	I-FID	train
RC-TAC-G	26.2	79.9	49.8	77.7	68.0	73.2	76.7	65.7
RC-TAC-C	23.1	81.2	41.6	66.1	87.2	48.4	108.2	40.9
CO-TAC	24.2	83.6	31.6	79.7	64.5	80.0	87.0	74.8
TAC-MENTOR	21.0	78.1	32.0	70.9	57.9	67.8	79.1	63.3

Table 4: CIFAR10 label-flip corruption Intra-FID and GAN-train results. Intra-FID: lower is better. GAN-train: higher is better.

Experiments: CIFAR10

Figure 2: Generated image previews for CIFAR10 with label-flipping corruption, corruption probability 0.45. Clockwise from top left: `RC-TAC-G`, `RC-TAC-U`, `CO-TAC`, `TAC-MENTOR`.

Conclusion

- While not perfect, the proposed methods (usually) outperform the baselines, and do not underperform relative to the baselines.
- They are able to do so whilst assuming nothing about the noise structure - not even that a transition matrix exists.

Future Work

- How do auxiliary classifiers and generators interact with each other in an auxiliary classifier GAN?
- Are there more compact ways to enhance GAN capabilities rather than introducing new networks into the ecosystem?
- What are the inherent properties of GANs?