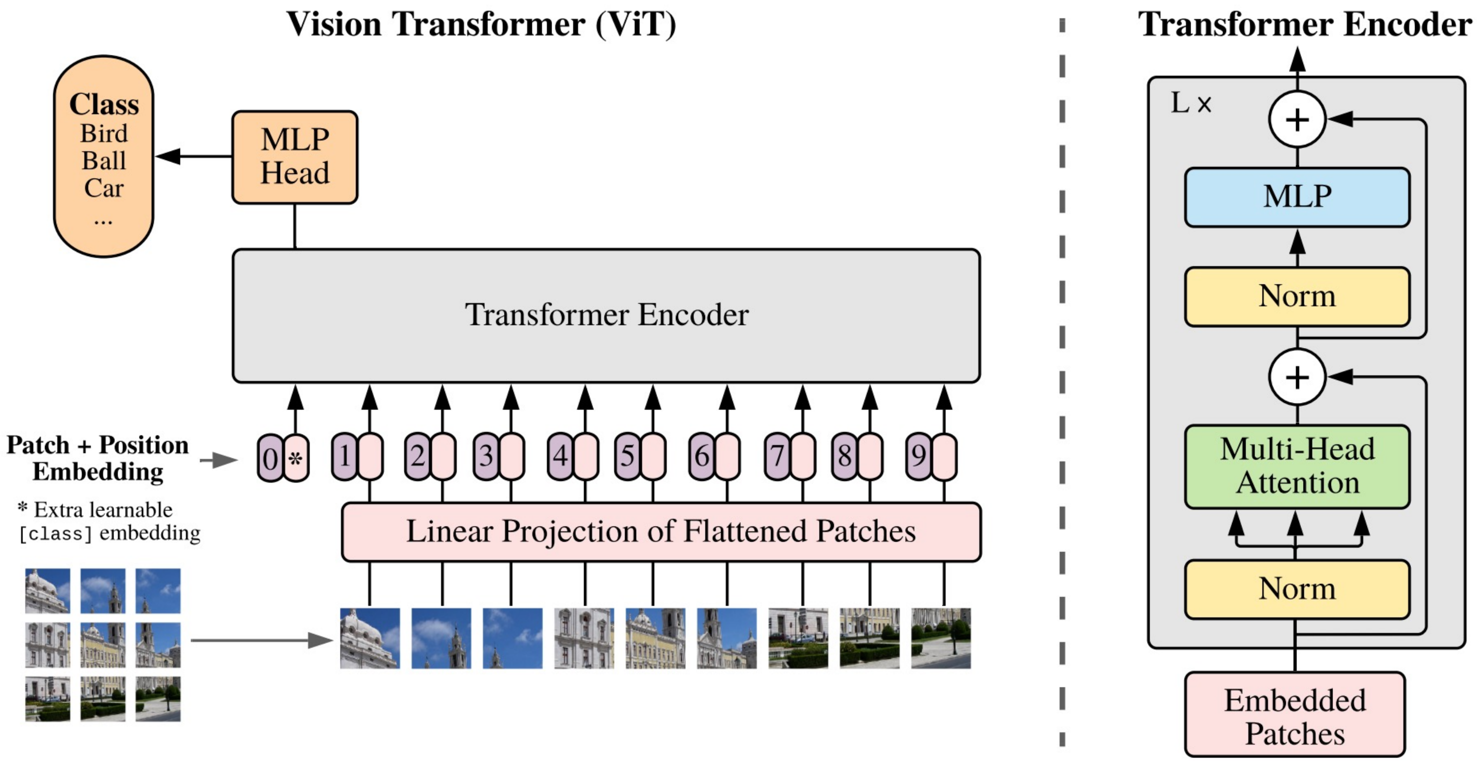
Implementation of Group Unlearning

23.12.22.
Jungyeon Koh

1. Usage of ViT for Extracting Features from Images

What is ViT (Vision Transformer)?



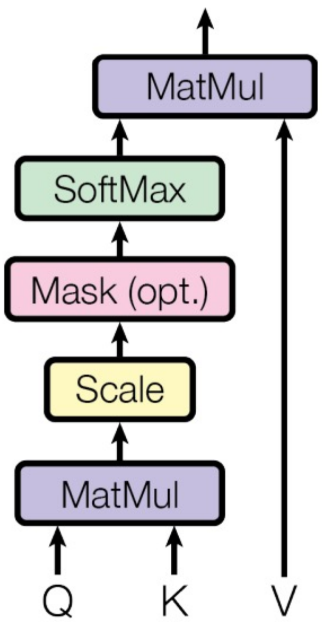
✓ ViT lacks **inductive bias** inherent to CNNs. Need enough data to be well-generalized.

Lack spatial relations between the input patches

* Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR 2021

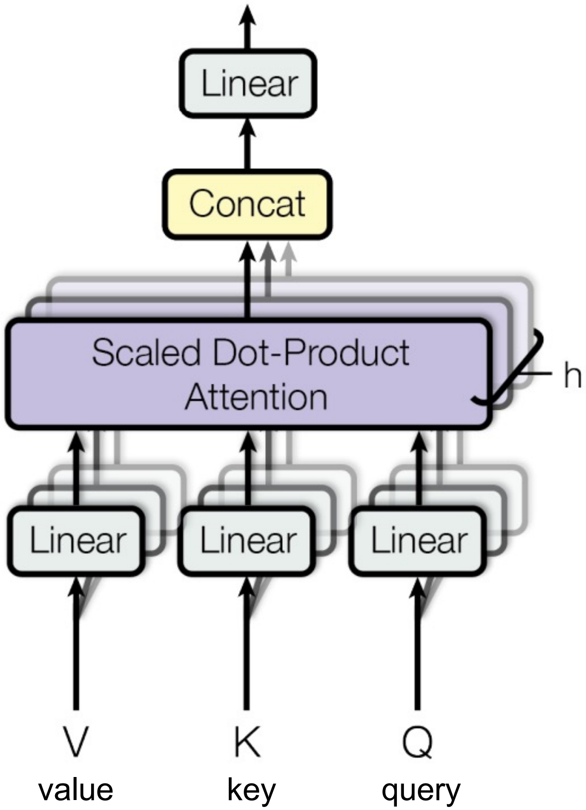
What is Multi-Head Attention?

Scaled Dot-Product Attention



$$\left(Attention(Q,K,V) = softmax\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V\right)^{h}$$

Multi-Head Attention

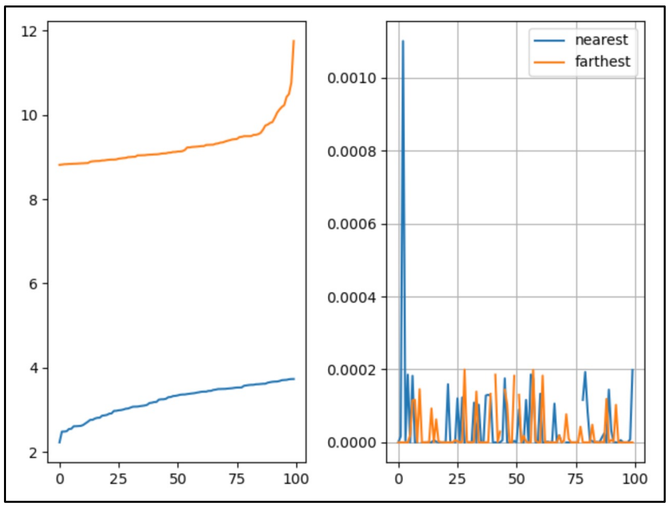


* Vaswani et al., “Attention Is All You Need”, NeurIPS 2017

ViT is better than CNN-based feature extractor

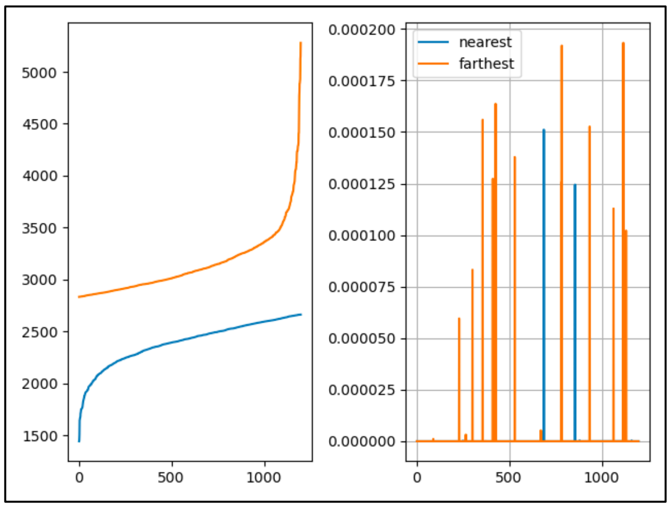
- ✓ Uses KNN to clump data points **within the same label** to capture subgroup coherence

- ✓ Need an assumption that **similar features have similar influences**
- ✓ Estimate l_2-norm of the influence function of the nearest and farthest neighbors



Nearest, Farthest $k = 100$ of 1000 samples

(Left) CNN



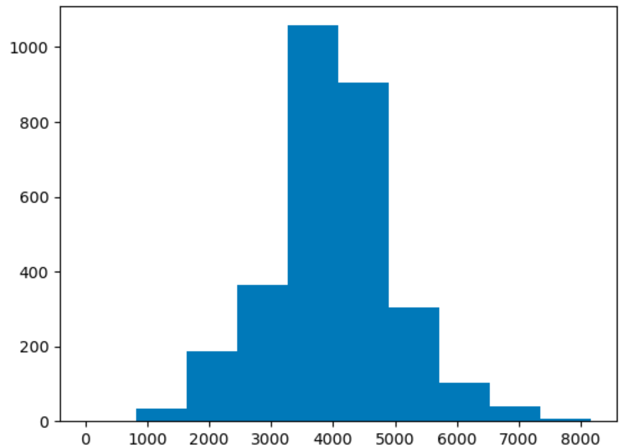
Nearest, Farthest $k = 1200$ of 3000 samples

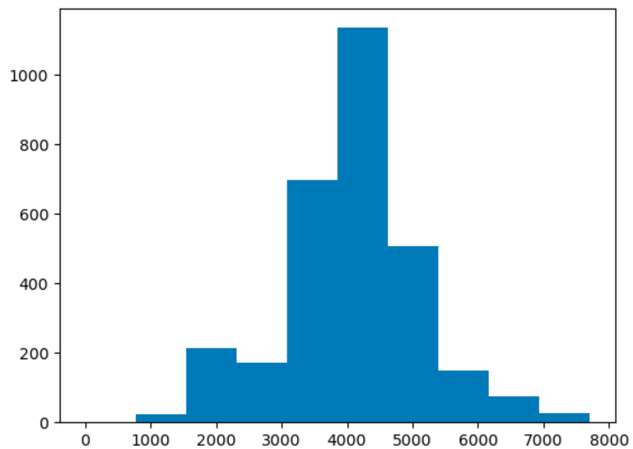
(Right) ViT

2. Composition of a priori Clusters

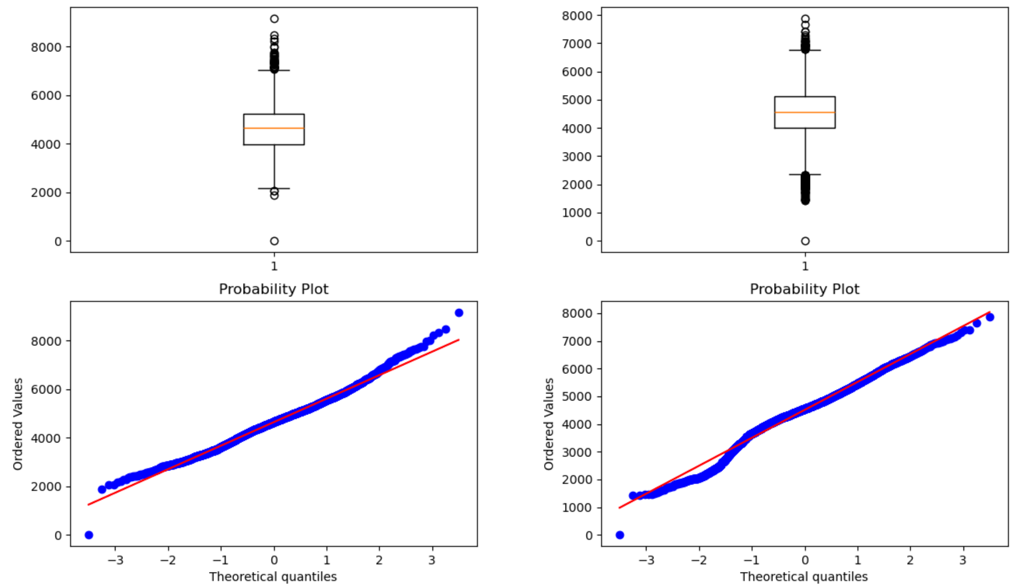
Normality test for feature distance

Feature distance shows a **bell-shaped** distribution





✓ **boxplot and probplot**



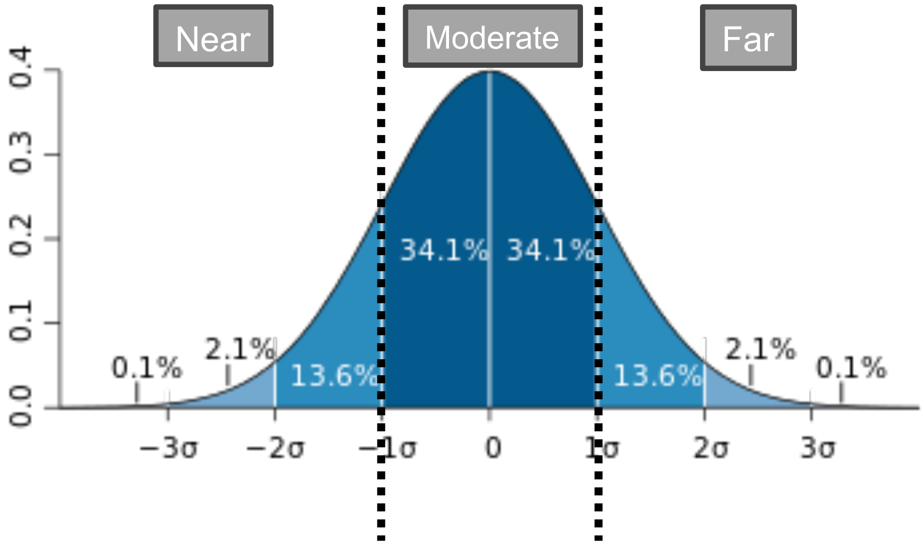
✓ **Shapiro-Wilk Test**

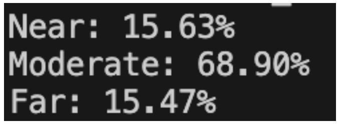
✓ **Normal Test: use skewness and kurtosis**

```
ShapiroResult(statistic=0.9753623008728027, pvalue=2.278858353776583e-22)
NormaltestResult(statistic=69.21552992201579, pvalue=9.333373517057439e-16)
```

✓ Anyway, let’s assume the data follows normal distribution

Group Unlearning Baseline 1: 3-steps

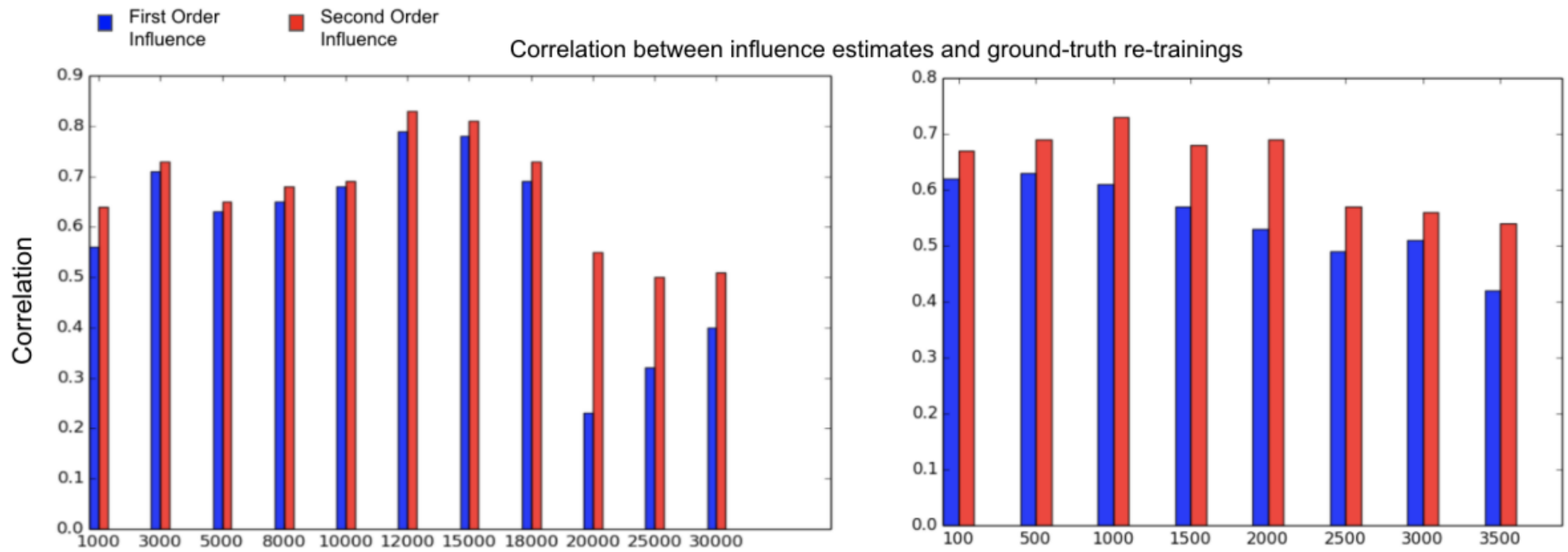




(Left) Theoretical (Right) Real

- ✓ Assume previous works cannot capture generality well. (∵ Group IF tends to underestimate the true effect.) ** Koh et al., NeurIPS 2019*
- ✓ Proposal:
  - ✓ Fixed # of steps: 3 (near, moderate, far)
  - ✓ In which order?
  - ✓ Or focus only on data within $\pm\sigma$ (aka moderate)
  - ✓ *Unsupervised learning*
  - ✓ *Cosine similarity*

Group Unlearning Baseline 2: $|U|$ and eval metric



Size of Group

* Basu et al., ICML 2020*
* (Left) random (Right) coherent; 100~3500 data from a specific class
* $|U|$ ranging from 1.6% to 60%

Table 1: Update results for the four selection criteria. Cross-entropy losses are computed on ResNet-18 with the MNIST dataset.

Criteria	Loss	Modification Ratio (MR%)			
		5	**10**	**30**	**50**
Top-k outputs	Self-loss ↑	6.23	6.21	**5.13**	4.89
	Test loss ↓	0.05	**0.11**	**0.64**	**0.69**
Top-k gradients	Self-loss ↑	**6.24**	**6.29**	4.95	**4.89**
	Test loss ↓	**0.04**	0.12	1.02	0.81
Threshold	Self-loss ↑	4.42	4.81	3.71	4.33
	Test loss ↓	0.09	0.65	1.78	1.18
Random	Self-loss ↑	4.42	4.79	3.36	4.26
	Test loss ↓	0.08	0.60	2.63	1.46

*Best: **bold**, second-best: underline.

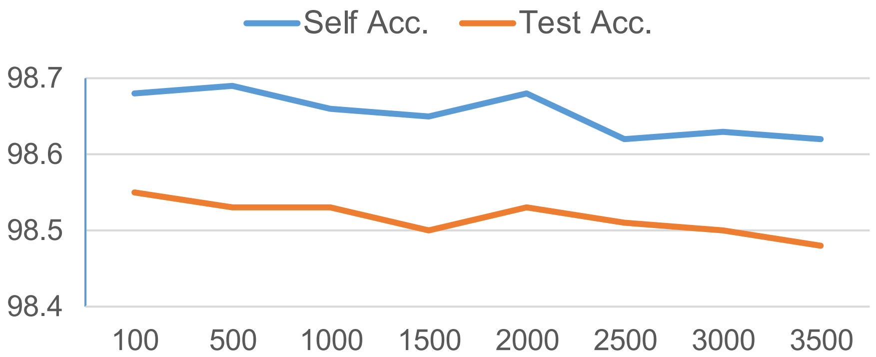
** Lyu et al., preprint*

3. Experiment Results

1) Coherent Unlearning
2) Random Unlearning

Performance Comparison: [1] coherent/retrain

- ✓ Mode: **coherent**
- ✓ Label: 0, 8
- ✓ Length of dataset: 5923, 5851
- ✓ (Original) Self accuracy: 98.97%
- ✓ (Original) Test accuracy: 98.77%
- ✓ Average over 50 experiments for each label



Size	100	500	1000	1500	2000	2500	3000	3500
Ratio (%)	1.6	8.4	16.9	25.3	33.8	42.2	50.7	59.1
Self Acc (%)	98.68	**98.65/ 98.73**	98.66	98.65	98.68	98.62	98.63	98.62
Test Acc (%)	98.55	**98.48/ 98.58**	98.53	98.50	98.53	98.51	98.50	98.48

- ✓ **Remarks**
  - ✓ For some cases, **inter-label variance is large**.
  - ✓ Self-accuracy is preserved well. (maybe model is simple)
  - ✓ Test-accuracy diminishes as the size of unlearned dataset grows.

Performance Comparison: [2] 1-step unlearning

- ✓ Mode: **coherent**
- ✓ Label: 0, 8
- ✓ Length of dataset: 5923, 5851
- ✓ (Original) Self accuracy: 98.97%
- ✓ (Original) Test accuracy: 98.77%

Size	100	500	1000	1500	2000	2500	3000	3500
Ratio (%)	1.6	8.4	16.9	25.3	33.8	42.2	50.7	59.1
Self Acc (%)								
Test Acc (%)								

Performance Comparison: [3] 3-step unlearning

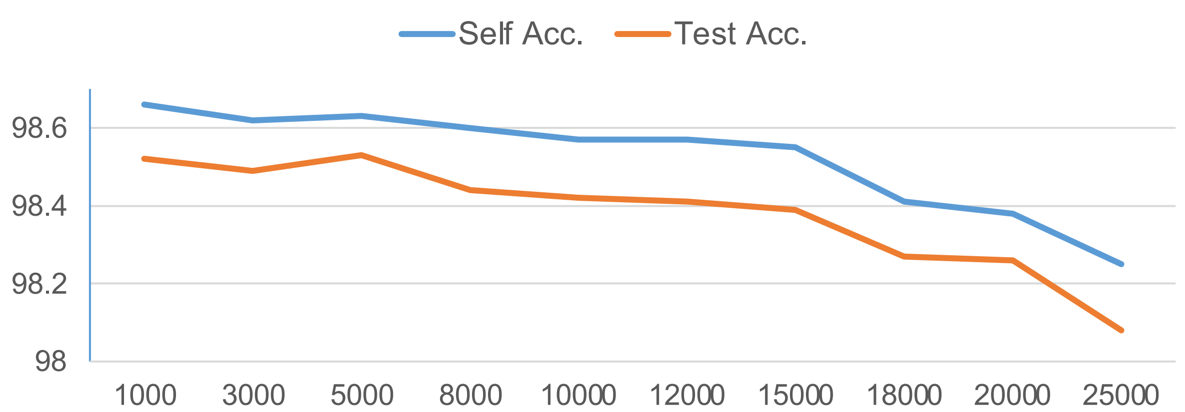
- ✓ Mode: **coherent**
- ✓ Label: 0, 8
- ✓ Length of dataset: 5923, 5851
- ✓ (Original) Self accuracy: 98.97%
- ✓ (Original) Test accuracy: 98.77%

Size	100	500	1000	1500	2000	2500	3000	3500
Ratio (%)	1.6	8.4	16.9	25.3	33.8	42.2	50.7	59.1
Self Acc (%)								
Test Acc (%)								

Performance Comparison: [1] random/retrain

- ✓ Mode: **random**
- ✓ Length of dataset: 55000
- ✓ (Original) Self accuracy: 98.97%
- ✓ (Original) Test accuracy: 98.77%
- ✓ Average over 50 experiments

Size	1000	3000	5000	8000	10000	12000	15000	18000	20000	25000
Ratio (%)	1.8	5.5	9.1	14.6	18.2	21.8	27.3	32.7	36.4	45.5
Self Acc (%)	98.66	98.62	98.63	98.60	98.57	98.57	98.55	98.41	98.38	98.25
Test Acc (%)	98.52	98.49	98.53	98.44	98.42	98.41	98.39	98.27	98.26	98.08



TODO

- ✓ Incomplete training
- ✓ GPU out of memory
- ✓ Scaling factor
- ✓ More sophisticated dataset and model…
- ✓ Comparing schemes: ~~Single-step~~ (Koh), Second-order (Basu), GIF (Lyu)…

```
scale_list = np.arange(1,20) *  1
for scale in scale_list:
    net = load_net(net, net_path)
    utils.update_network(net, influence / scale, index_list)

    print(f"PIF scale: {scale}")
    evaluate(net, corrupt_dataloader)
    evaluate(net, relabel_dataloader)
    evaluate(net, clean_dataloader, label)
    evaluate(net, test_dataloader)
    print("")
```