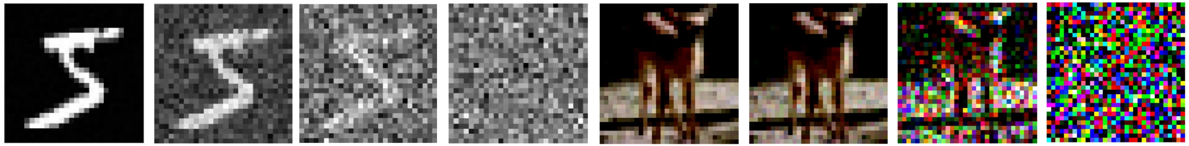


Team #19

Course Project
27/05/2024

Team Project on the course "Deep Learning"

Few Shot Generative Classification

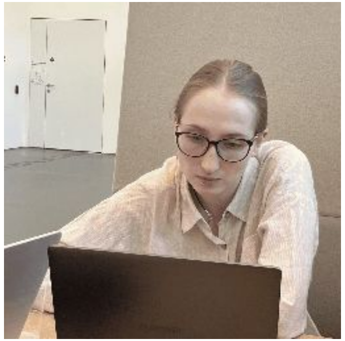
Kamil Garifullin
Ignat Melnikov
Artem Alekseev
Irina Lebedeva
Viktoriia Zinkovich

Our team #19

class = ‘spider’

Ignat Melnikov
Data Science, MS-1

class = ‘cat’



Irina Lebedeva
Internet of Things, MS-1

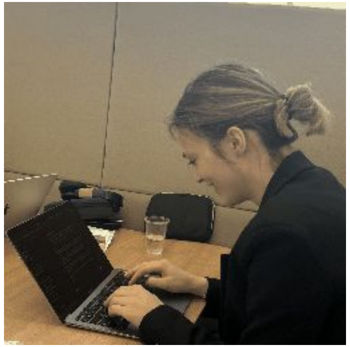
VAE team

class = ‘bird’

Kamil Garifullin
Data Science, MS-1

Diffusion team

class = ‘cat’

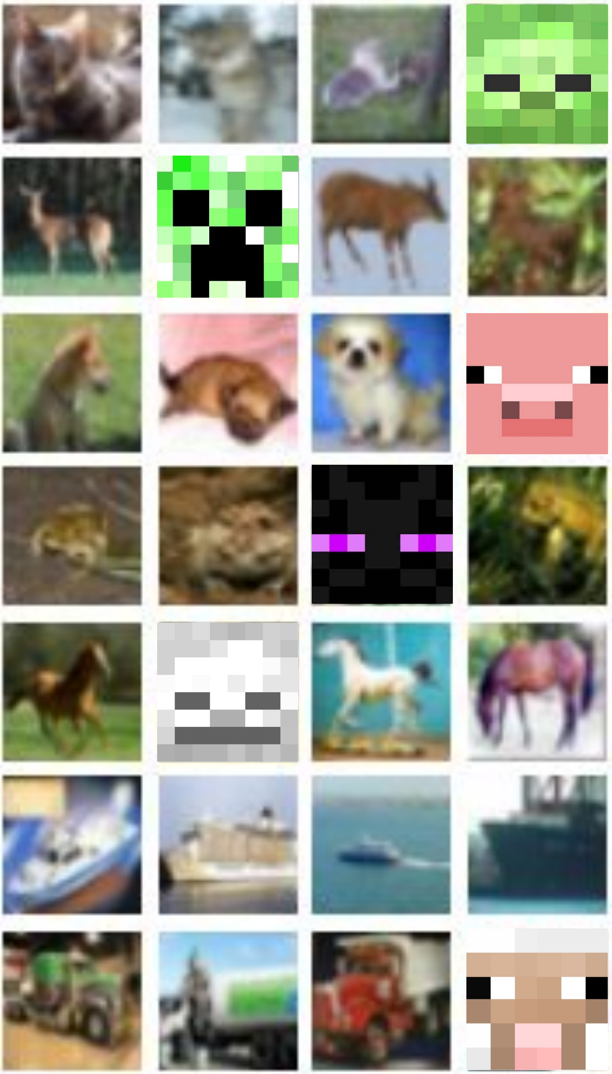


Viktoriia Zinkovich
Data Science, MS-1

class = ‘snowman’

Artem Alekseev
Data Science, MS-1

GANgsters team



Introduction

Motivation for our research, problem statement, overview of related works

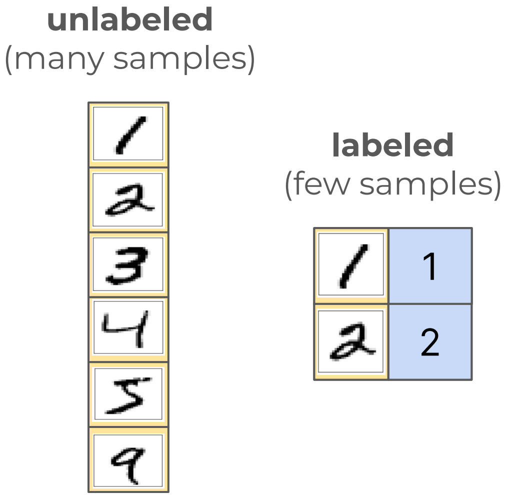
Problem

Traditional **supervised** classification approaches limits the scalability and efficiency of neural network training...

Time-consuming data collection of labeled images

Poorly labeled data

Significant **human efforts** and computational resources

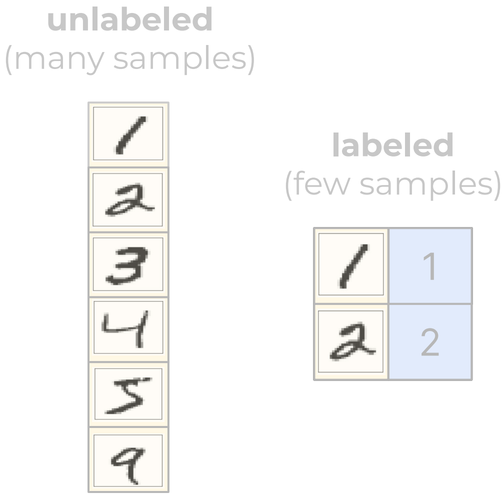


[1] **The Unreasonable Effectiveness of Deep Features as a Perceptual Metric.** R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang. 2018.

Problem

Traditional **supervised** classification approaches limits the scalability and efficiency of neural network training...

- **Time-consuming** data collection of labeled images
- **Poorly** labeled data
- Significant **human efforts** and computational resources



But **how to increase prediction accuracy** if labeled dataset is small?

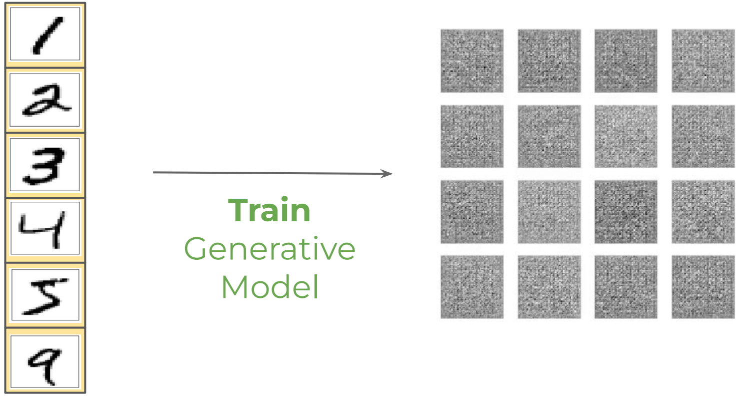
[1] **The Unreasonable Effectiveness of Deep Features as a Perceptual Metric.** R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang. 2018.

Motivation

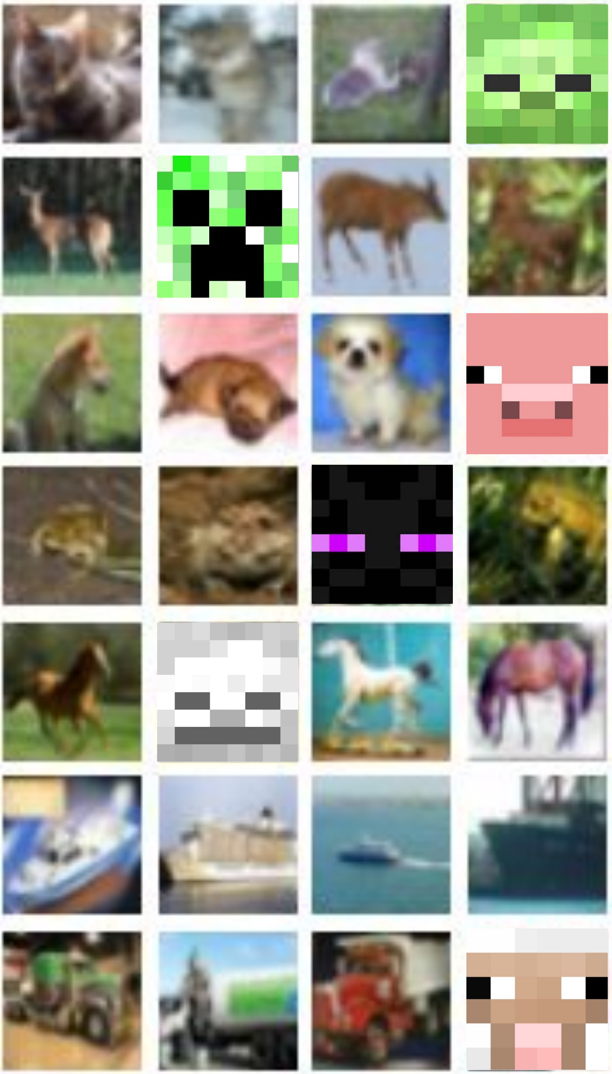
Implement **Generative Models** for Classification Tasks...

Potential of generative models to **alleviate the need for manual annotation** by learning feature representations directly from unlabeled data

unlabeled dataset
(many samples)



[2] **Structured Denoising Diffusion Models in Discrete State-Spaces.** J. Austin, D.D. Johnson, J. Ho, D. Tarlow, R. van den Berg. 2023.



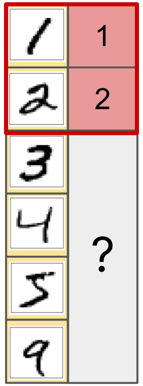
Methods

general idea

idea behind our method and pipeline of the work

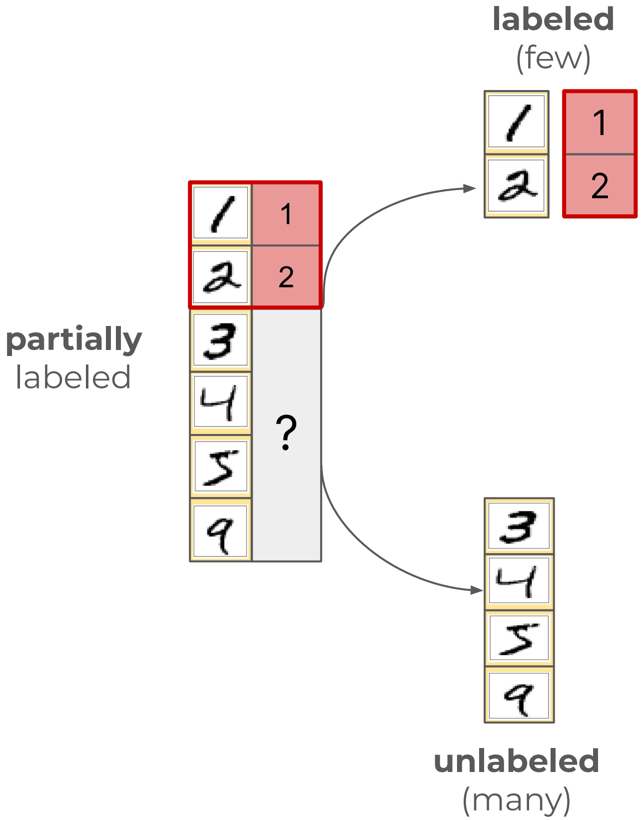
Concept

partially labeled

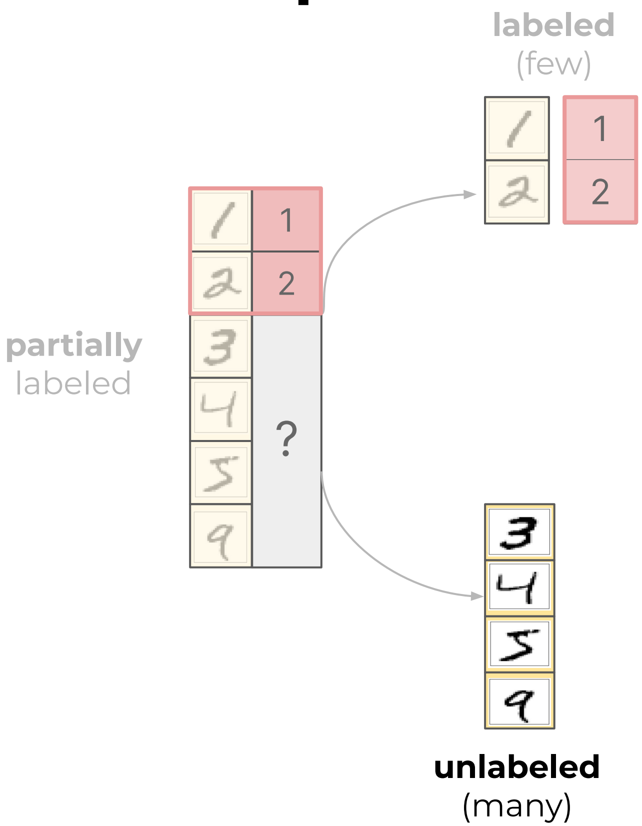


we have a **small labeled** dataset and a **large unlabeled** dataset ...

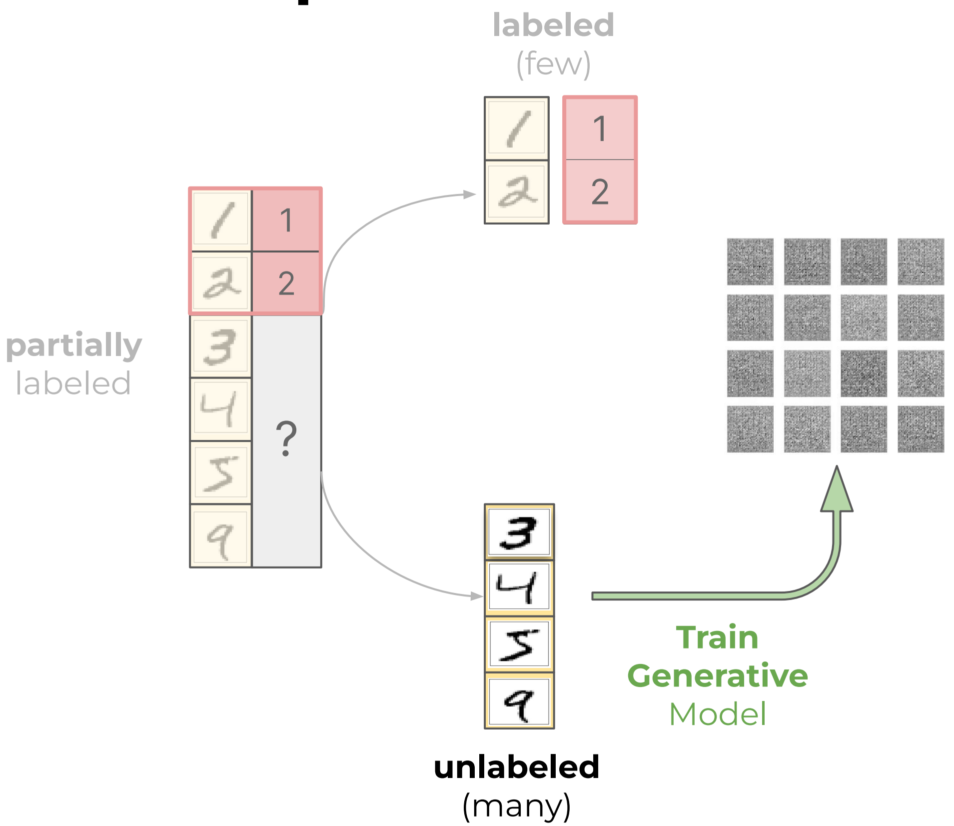
Concept



Concept

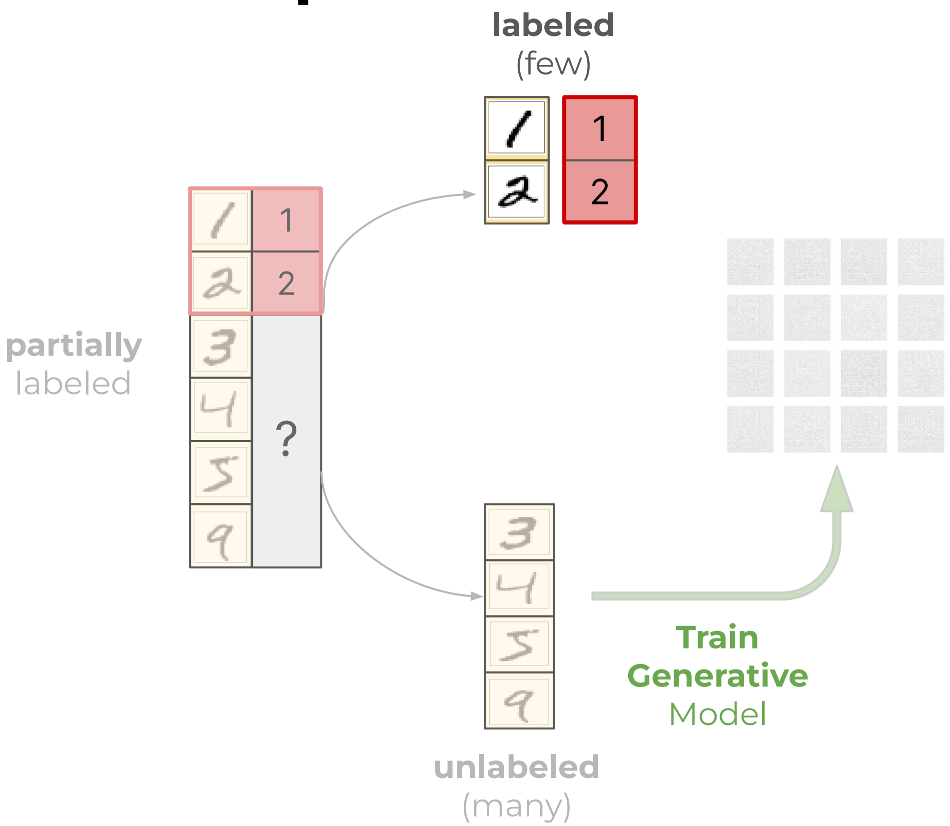


Concept



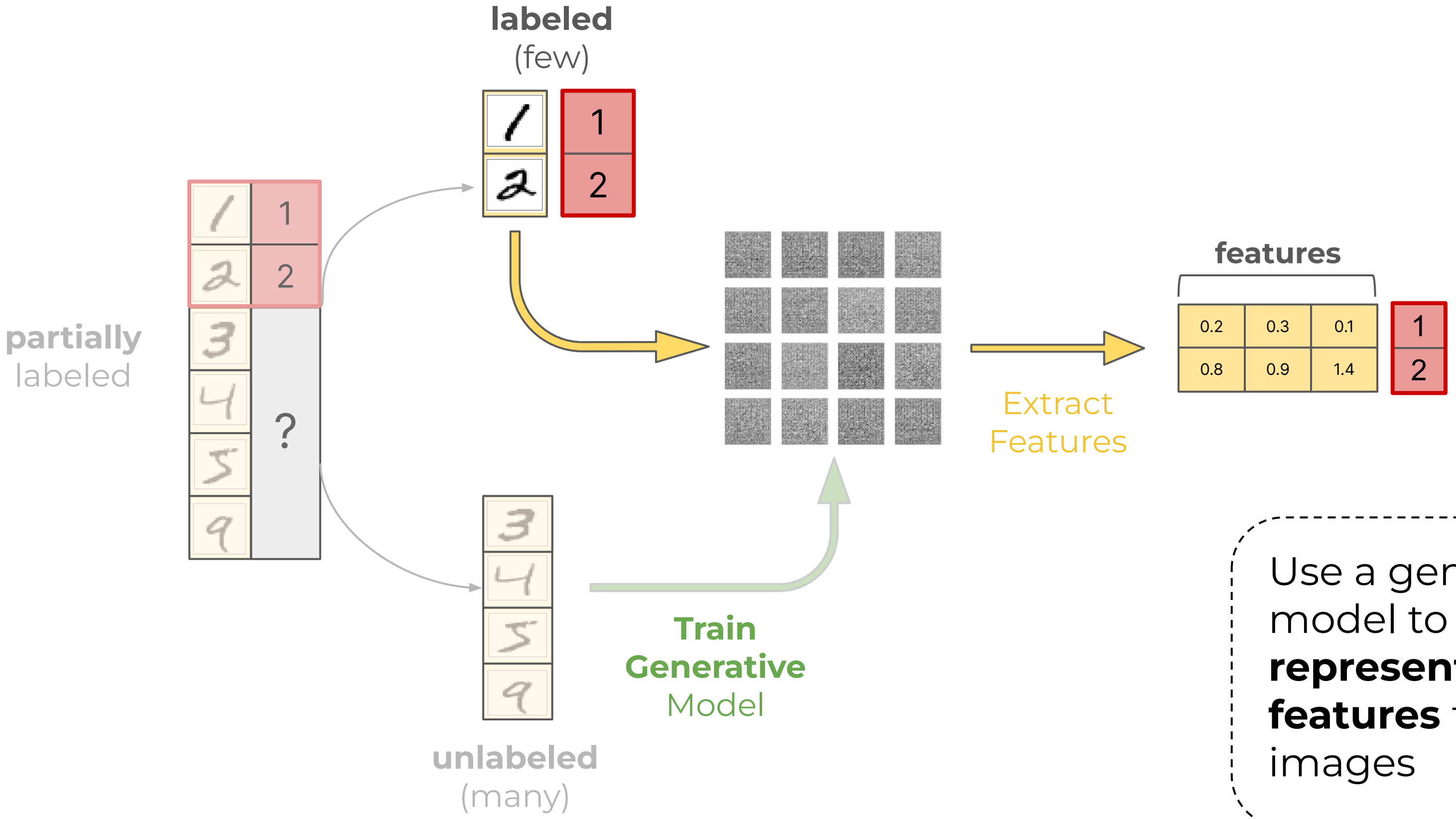
Training a generative model on **unlabeled** data

Concept



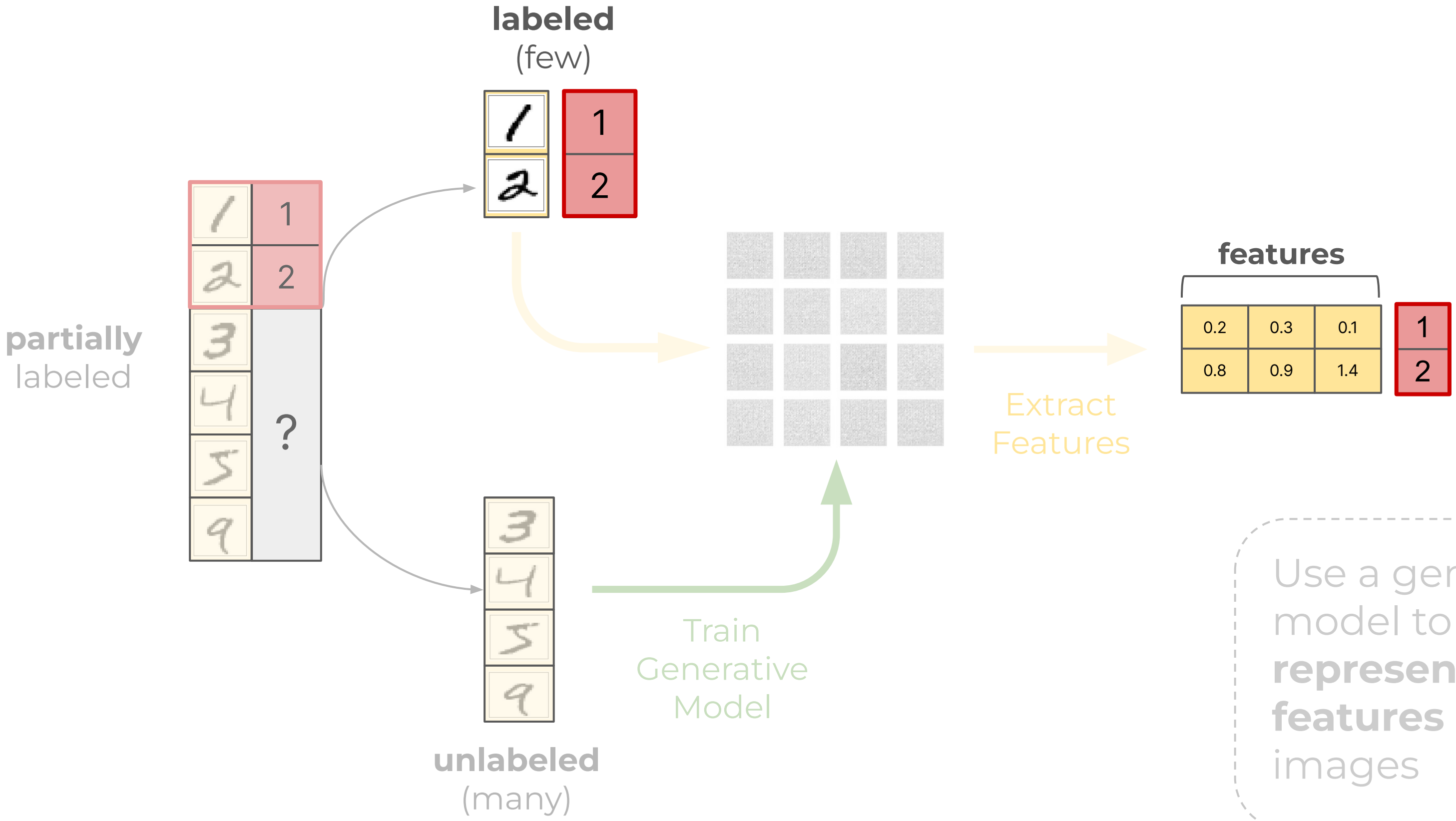
Then we work with **small labeled** dataset!

Concept



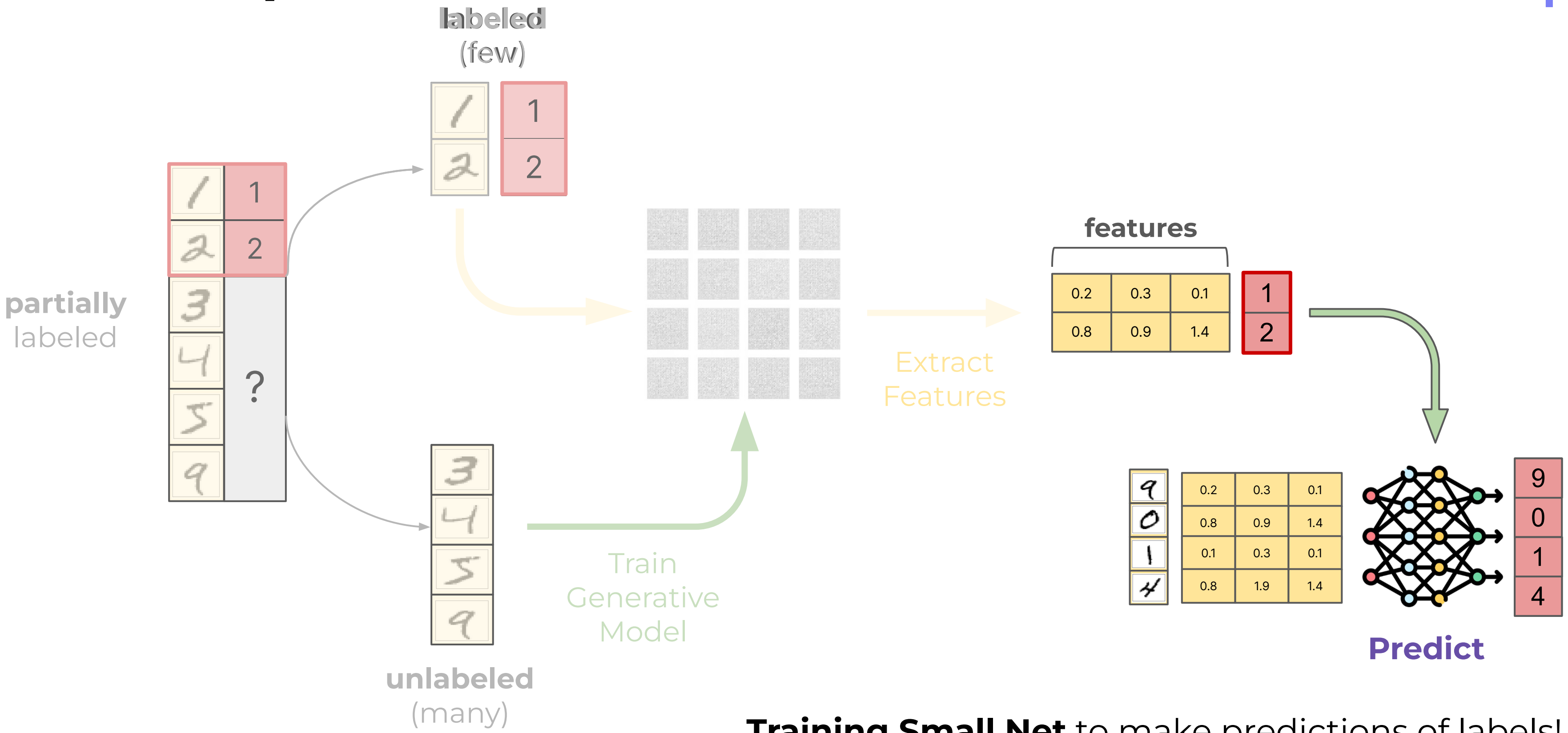
Use a generative model to **extract more representative features** for labeled images

Concept

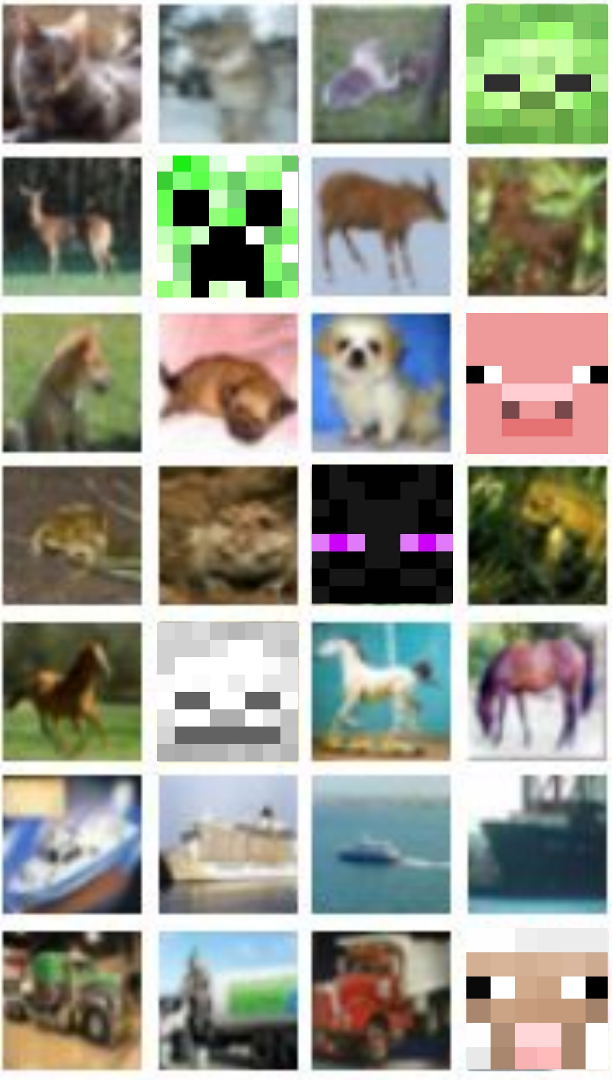


Use a generative model to **extract more representative features** for labeled images

Concept



Training Small Net to make predictions of labels!



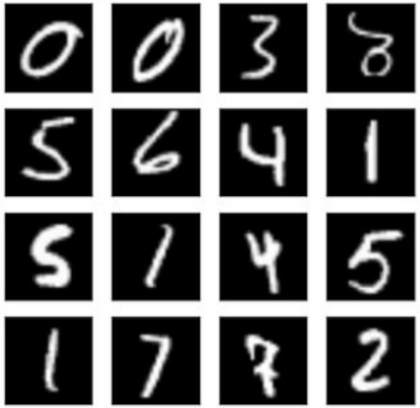
Datasets

description of the data used in the following research

Datasets

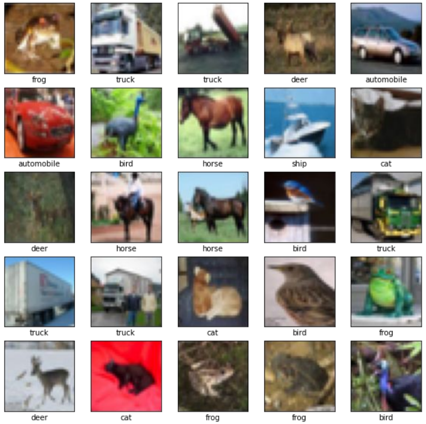
Characteristic of the used data for training

MNIST



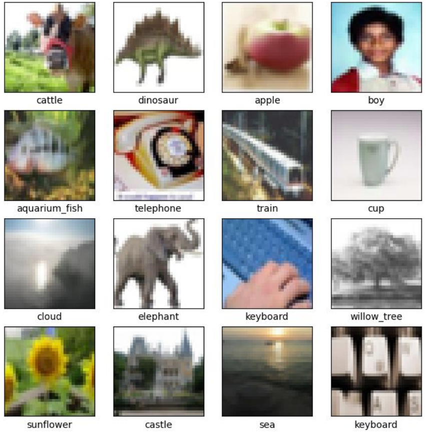
Shape: 28x28
Train set: 60000
Test set: 10000
Classes: 10

CIFAR-10

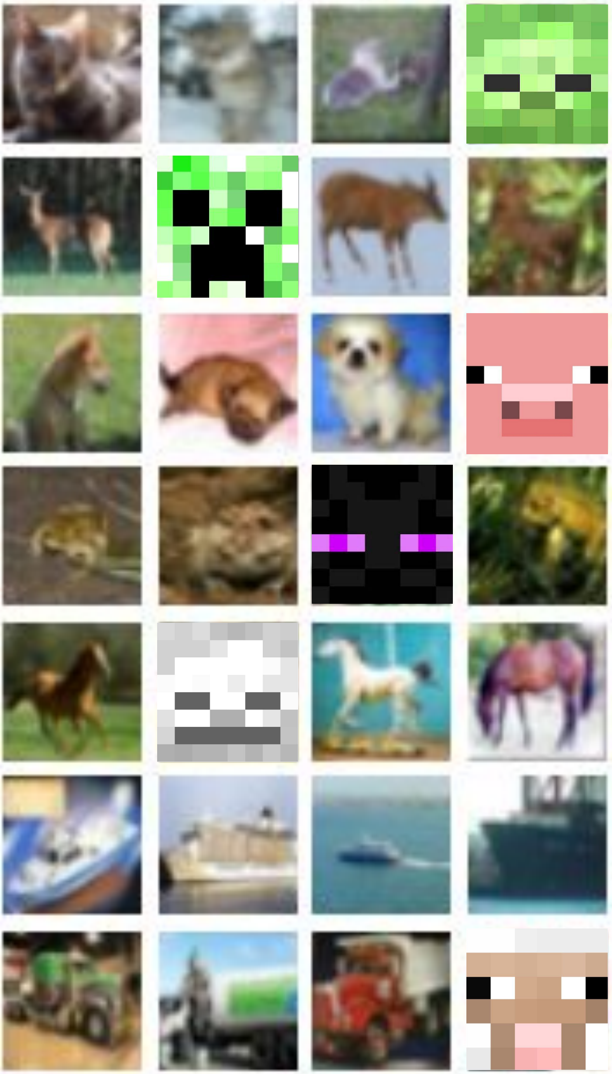


Shape: 32x32
Train set: 50000
Test set: 10000
Classes: 10

CIFAR-100



Shape: 32x32
Train set: 50000
Test set: 10000
Classes: 100



Methods

feature extraction

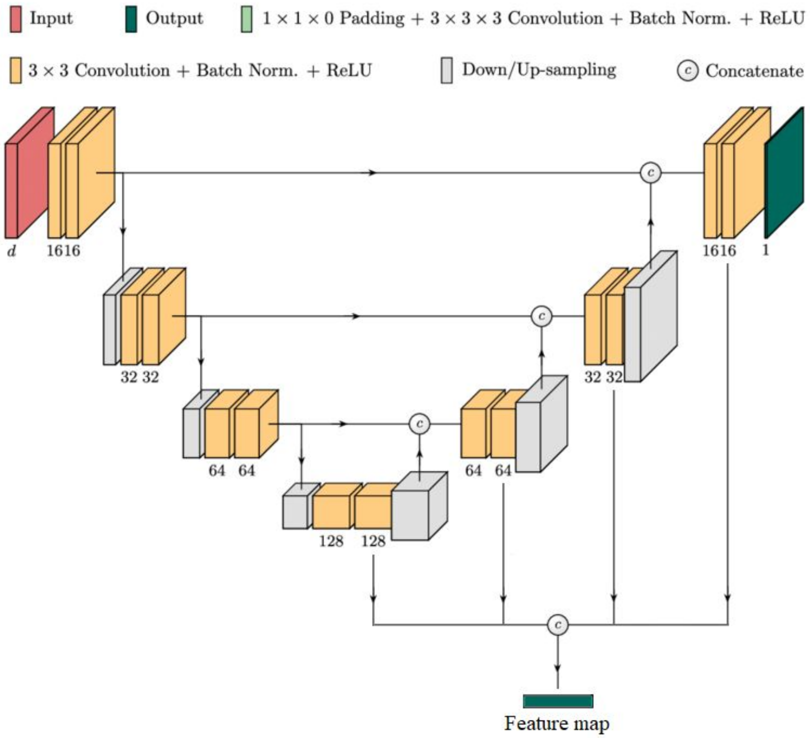
investigation of **generative model architectures** and explanation of how we extract features from them to train small neural network

Diffusion Model

How we **extract features from diffusion model** to train on them small neural network for classification:

1. As an architecture, **Residual UNet** was used

2. Select **features from last layers** of net and concatenate them in one array

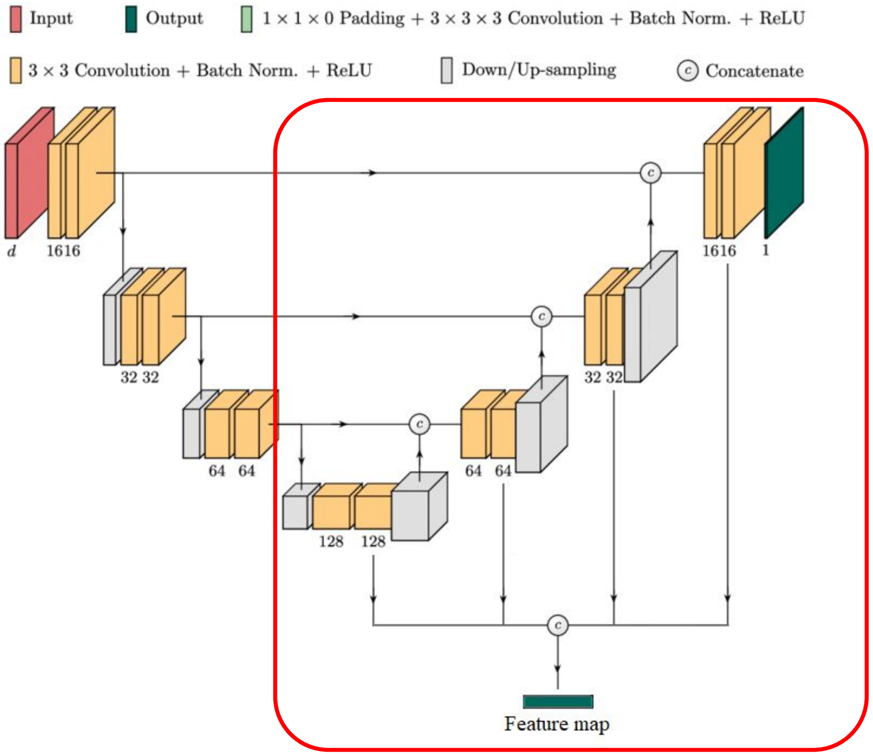


Diffusion Model

How we **extract features from diffusion model** to train on them small neural network for classification:

1. As an architecture, **Residual UNet** was used

2. Select **features from last layers** of net and concatenate them in one array

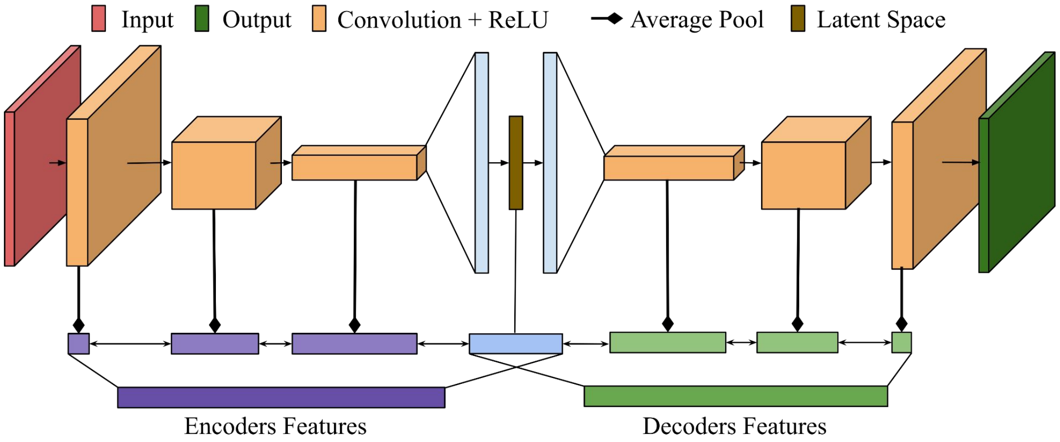


Variational AutoEncoder

Select **features from different layers** of net and concatenate them in one array

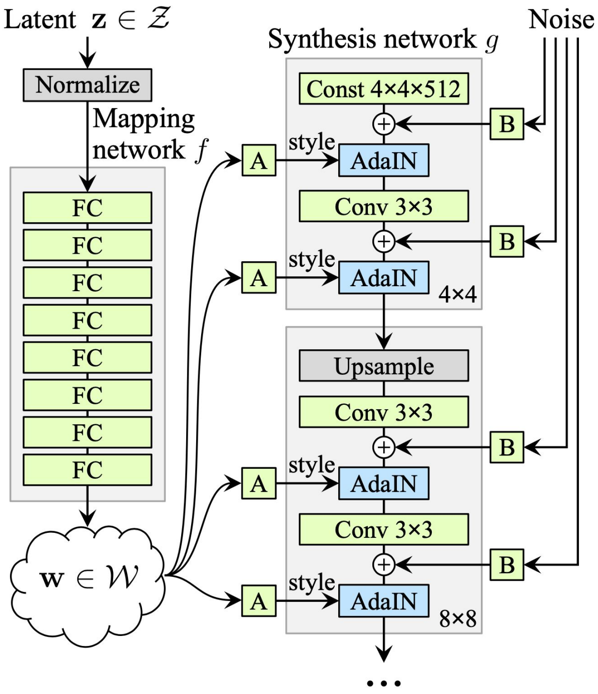
Results in three models:

1. encoder
2. decoder
3. stacked



Generative Adversarial Network

GAN imitation: common GAN is **non-invertible**

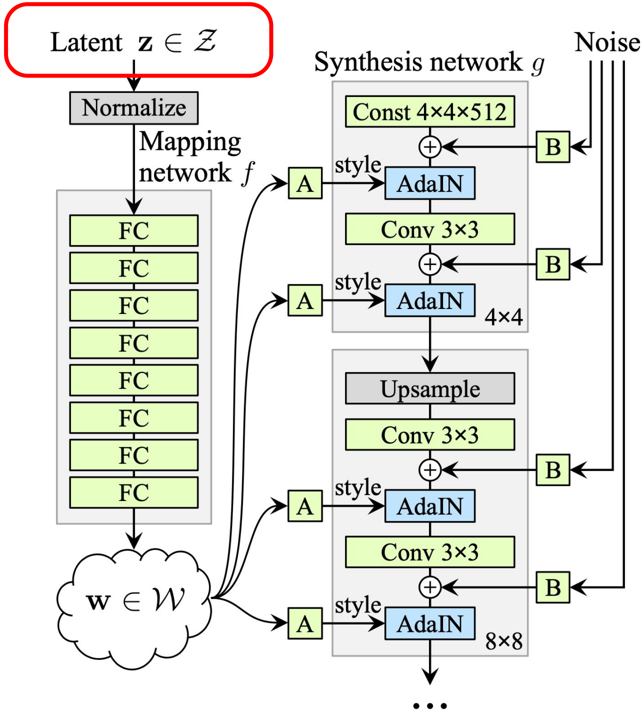


[1] **A Style-Based Generator Architecture for Generative Adversarial Networks.** T. Karras, S. Laine, T. Aila, 2019.

Generative Adversarial Network

GAN imitation: common GAN is **non-invertible**

can **only generate images from random noise** and cannot extract embeddings from real images



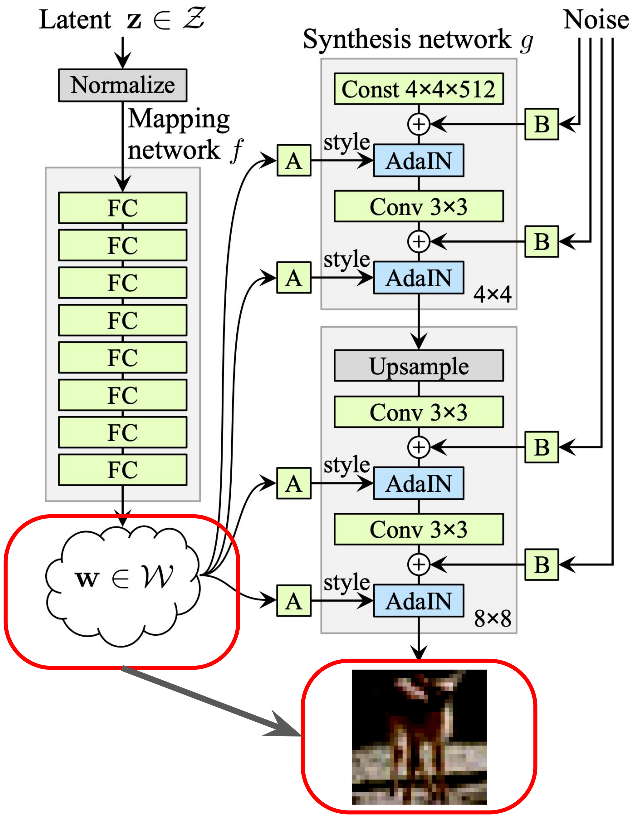
[1] **A Style-Based Generator Architecture for Generative Adversarial Networks.** T. Karras, S. Laine, T. Aila, 2019.

Generative Adversarial Network

GAN imitation: common GAN is
non-invertible

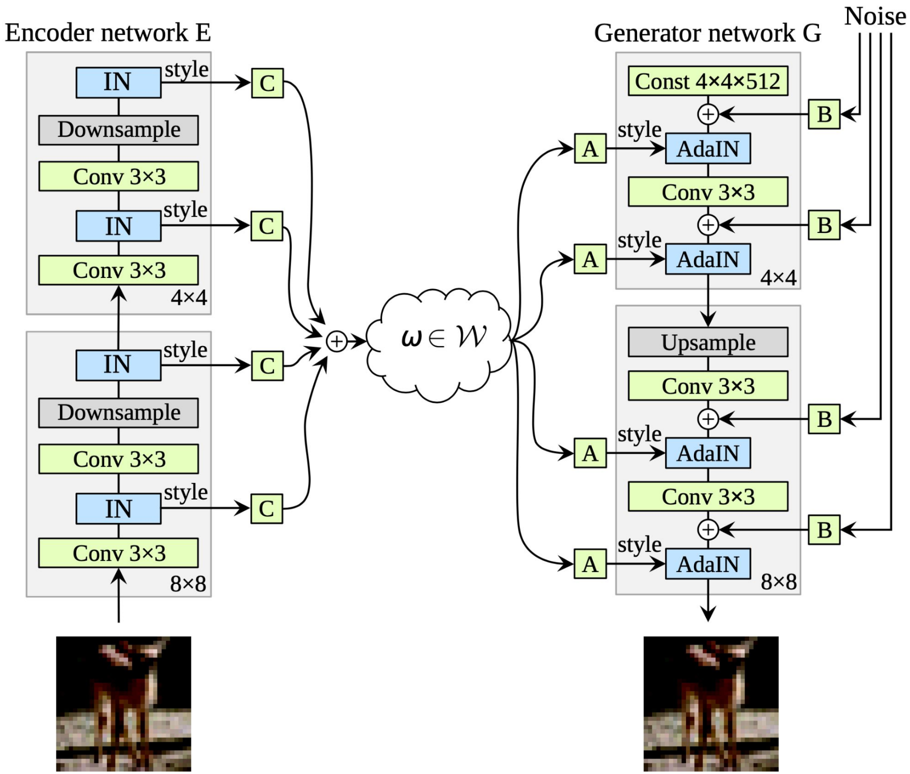
can **only generate images from random noise** and cannot extract embeddings from real images

Vectors from
latent space!



Generative Adversarial Network

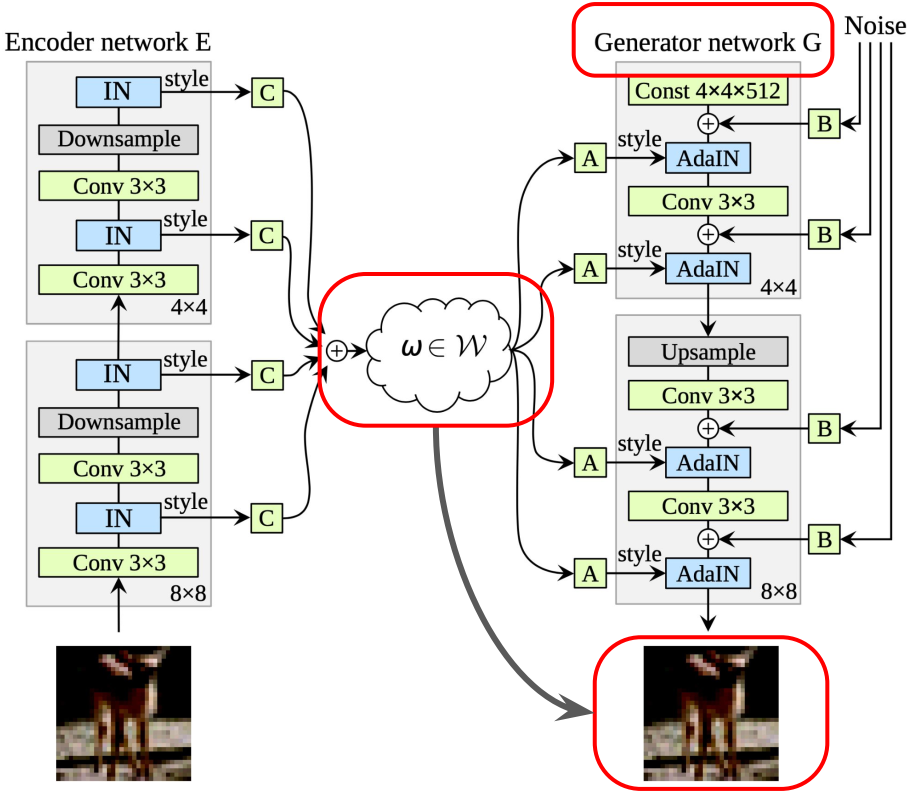
train an encoder along with the generator and discriminator!



Generative Adversarial Network

train an encoder along with the generator and discriminator!

generator:
vector –→ image

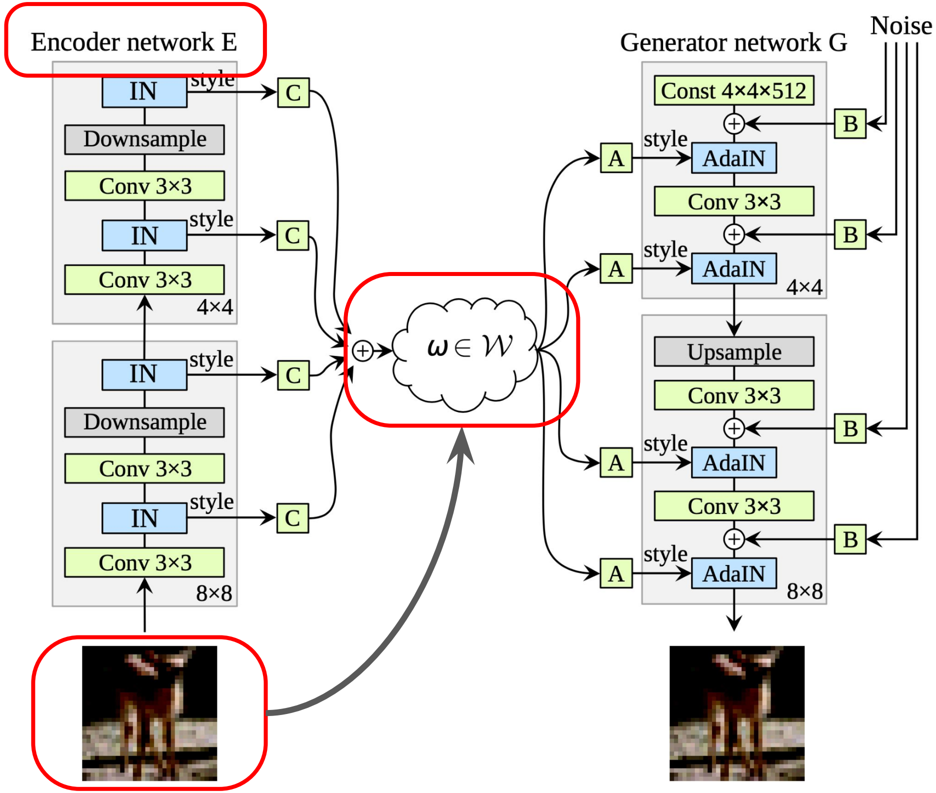


Generative Adversarial Network

train an encoder along with the generator and discriminator!

generator:
vector –→ image

encoder:
Image –→ vector



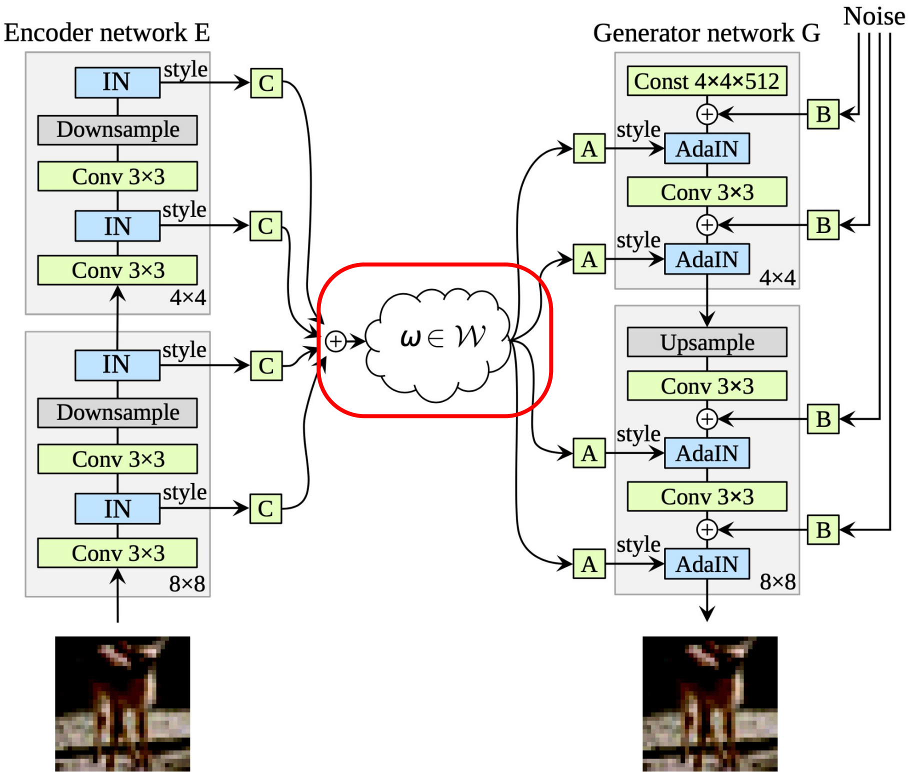
Generative Adversarial Network

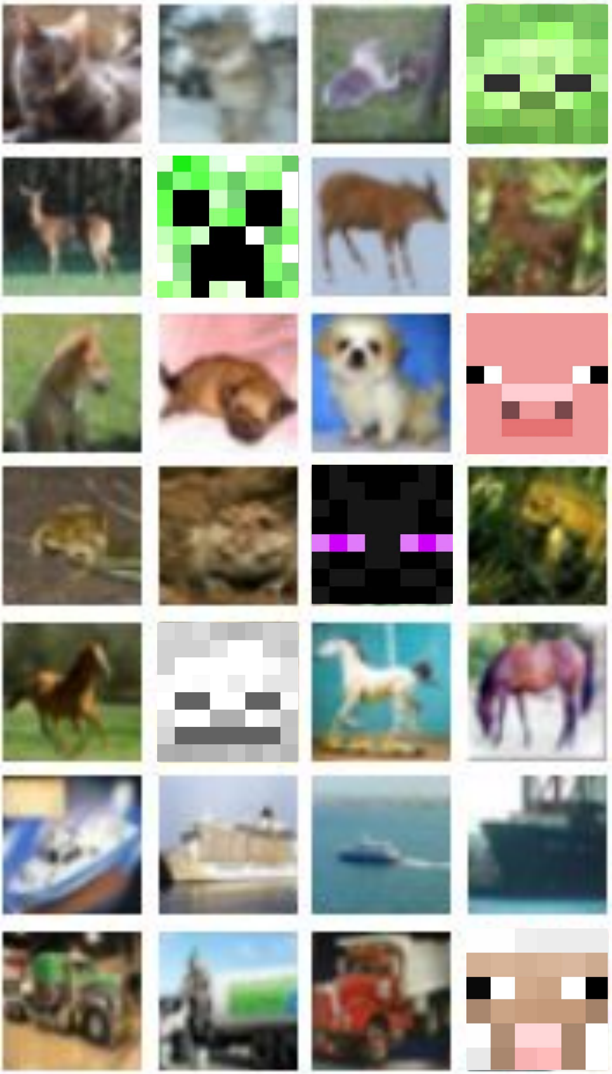
train an encoder along with the generator and discriminator!

generator:
vector –→ image

encoder:
Image –→ vector

w = our features!





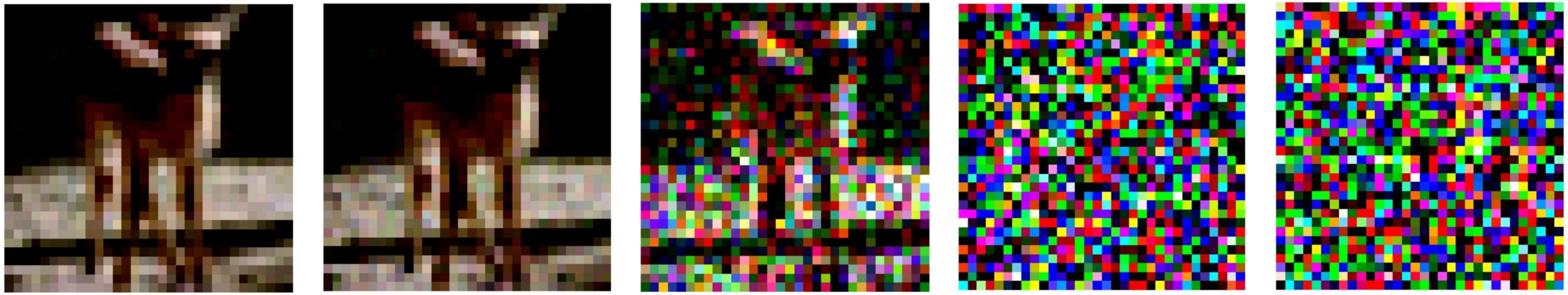
Hyperparameters

for diffusion model

Influence of choice of diffusion noise degree in images on model performance

Diffusion Model

Select features from different steps of forward diffusion process, diffusion noise step is a **hyperparameter**

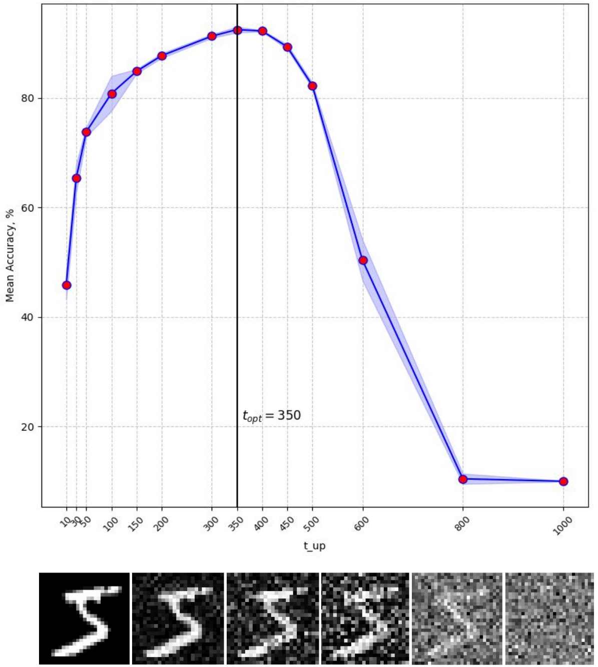


1/1000 10/1000 100/1000 500/1000 900/1000

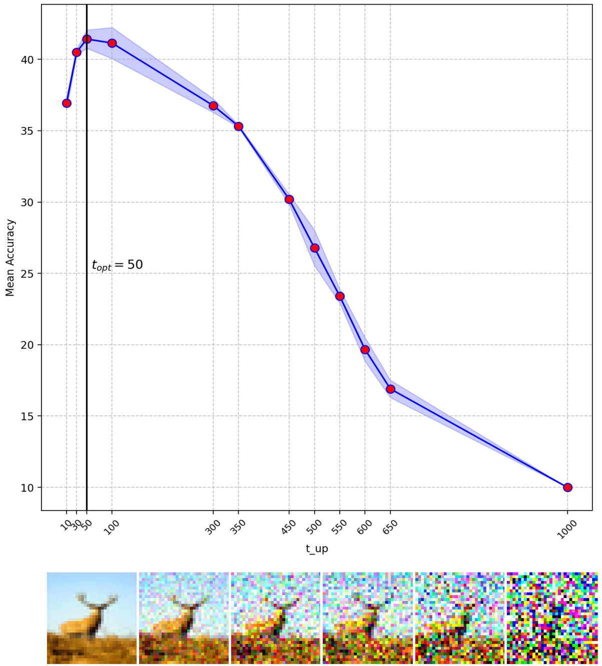
t_noise

Diffusion Model

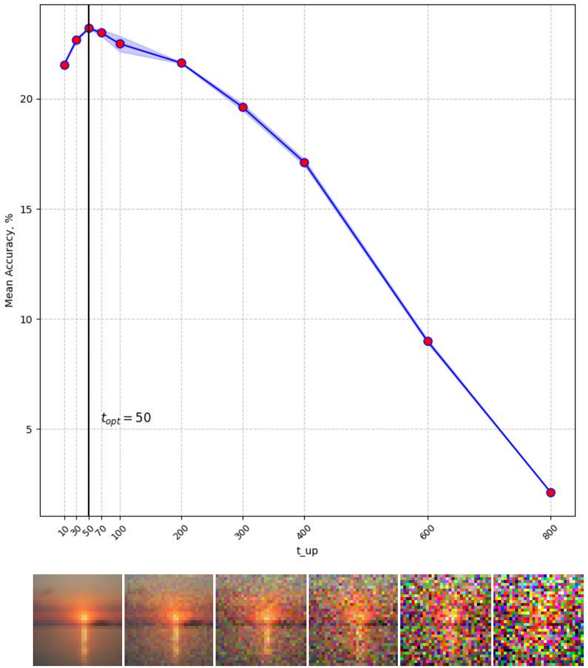
MNIST: $t_{opt} = 350$



CIFAR-10: $t_{opt} = 50$

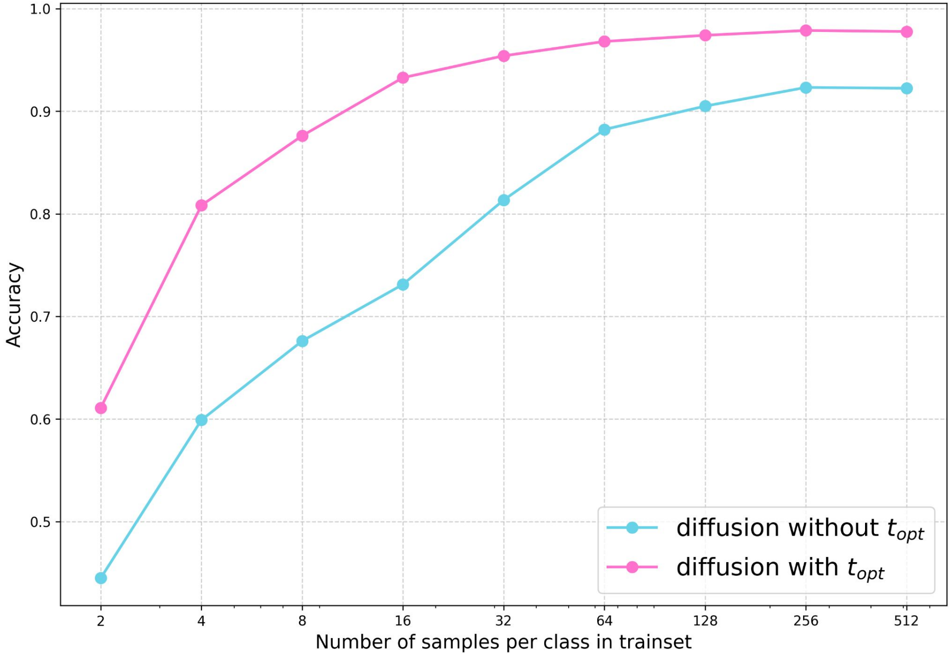


CIFAR-100: $t_{opt} = 50$



Diffusion Model

Selecting pictures from the optimal noise step improved the performance of the diffusion model

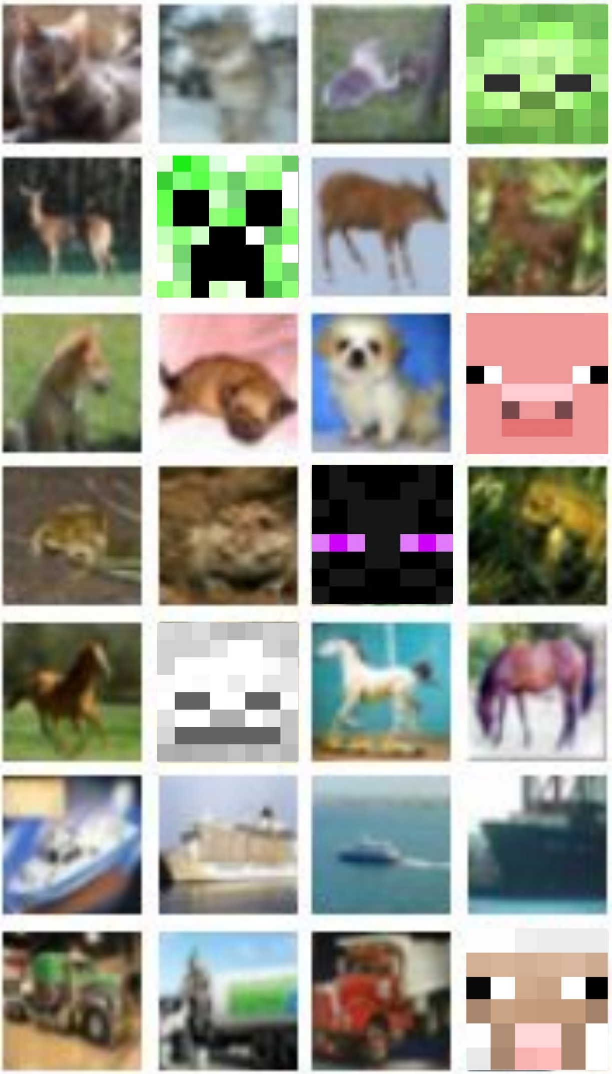


dataset size = 4

60% → 81%
before after

dataset size = 16

73% → 92%
before after



Results

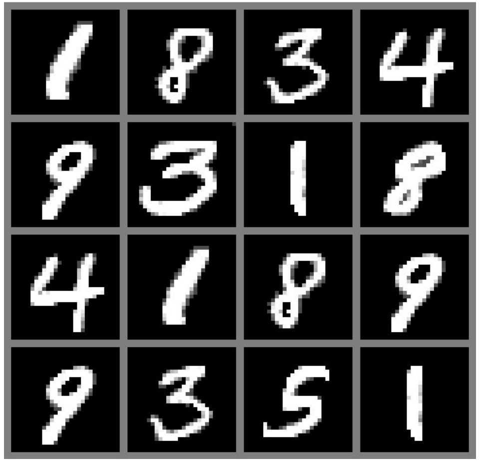
MNIST

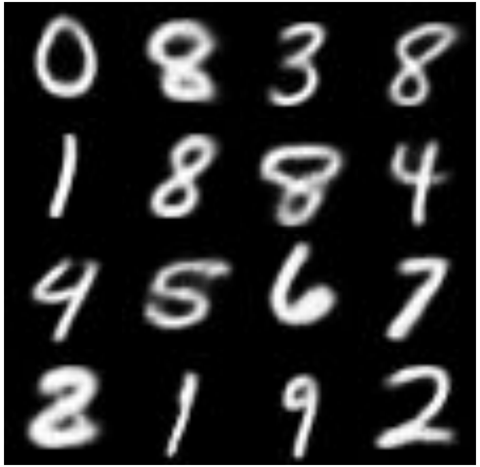
Diffusion, VAE, GAN models results and comparison of their performance with resnet-18

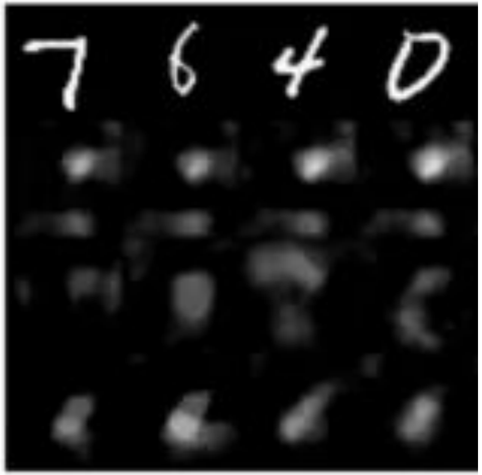
Generation quality

Visual estimation of generation quality, all models were trained on T4 GPU

diffusion	VAE	GAN
time ~ 40 min	time ~ 5 min	time ~ 5 h



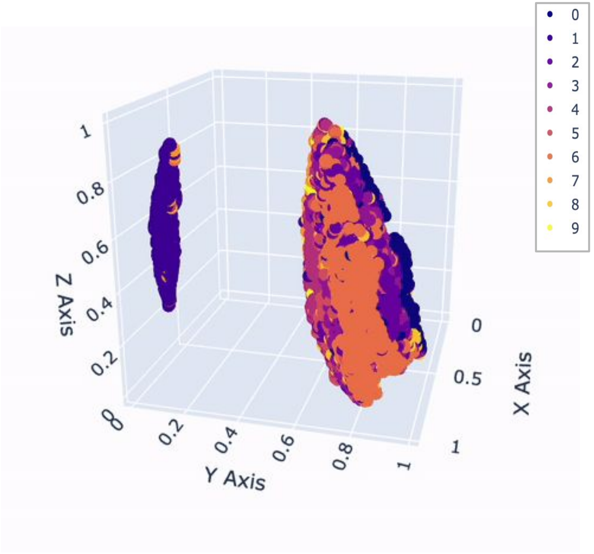




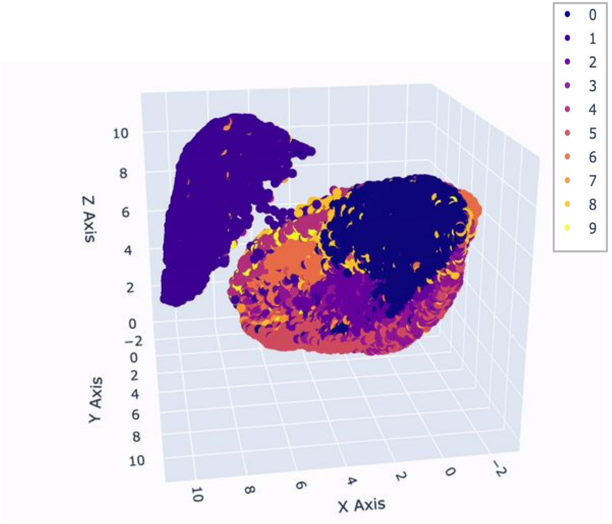
UMAP projection

Projection of features after training on **unlabeled** dataset, **MNIST dataset**

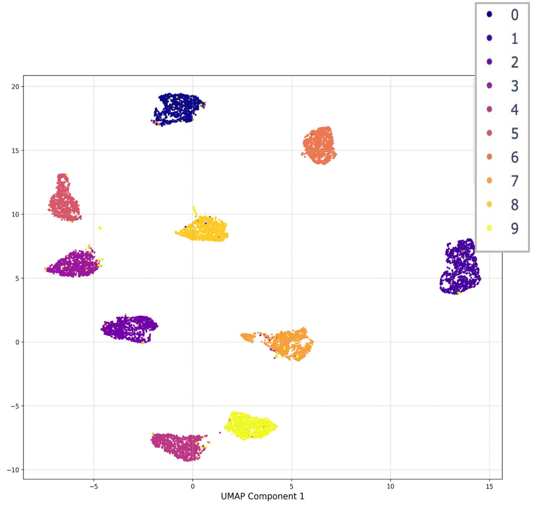
diffusion
feature dim = 512



VAE *(encoder)*
feature dim = 236

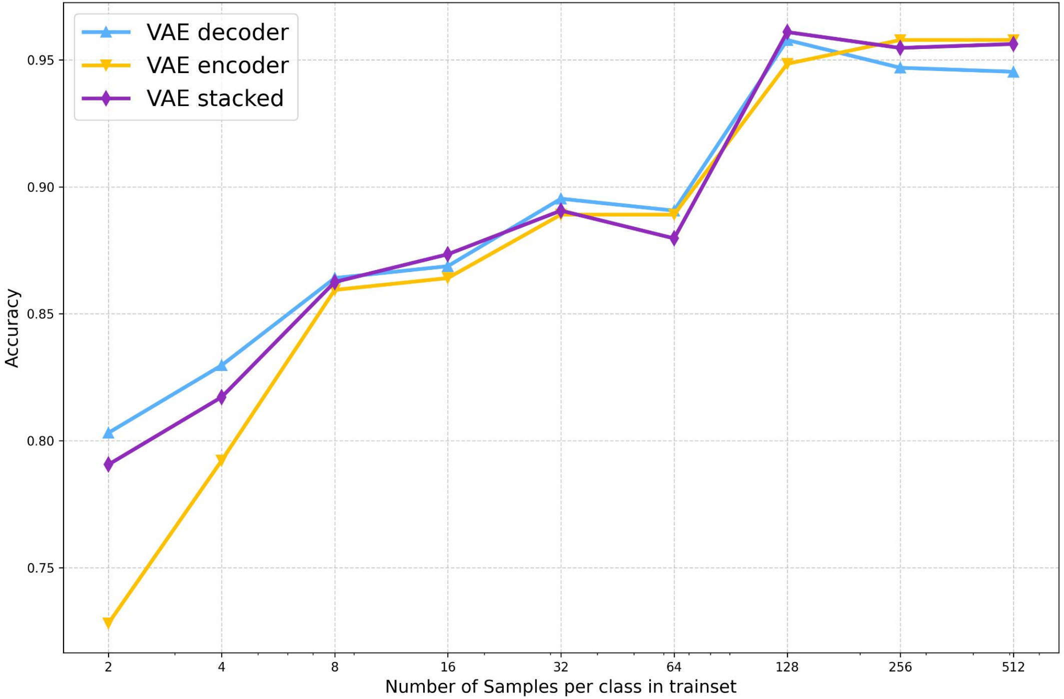


GAN
feature dim = 128



Training Results for MNIST

Three different ways of feature extraction from VAE are compared:

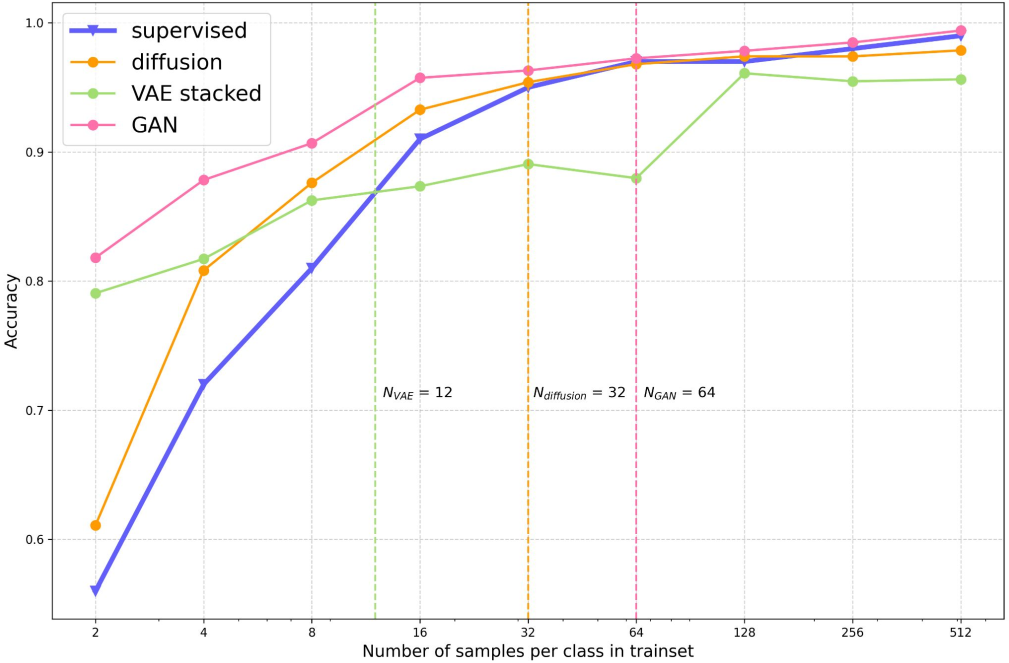


1. there is practically **no difference** in which way to extract features

2. **“stacked”** method is slightly better

Training Results for MNIST

Training a **nonlinear** (Linear+ReLU+Linear) model on features extracted from generative models



dataset sizes at which our models outperform ResNet-18:

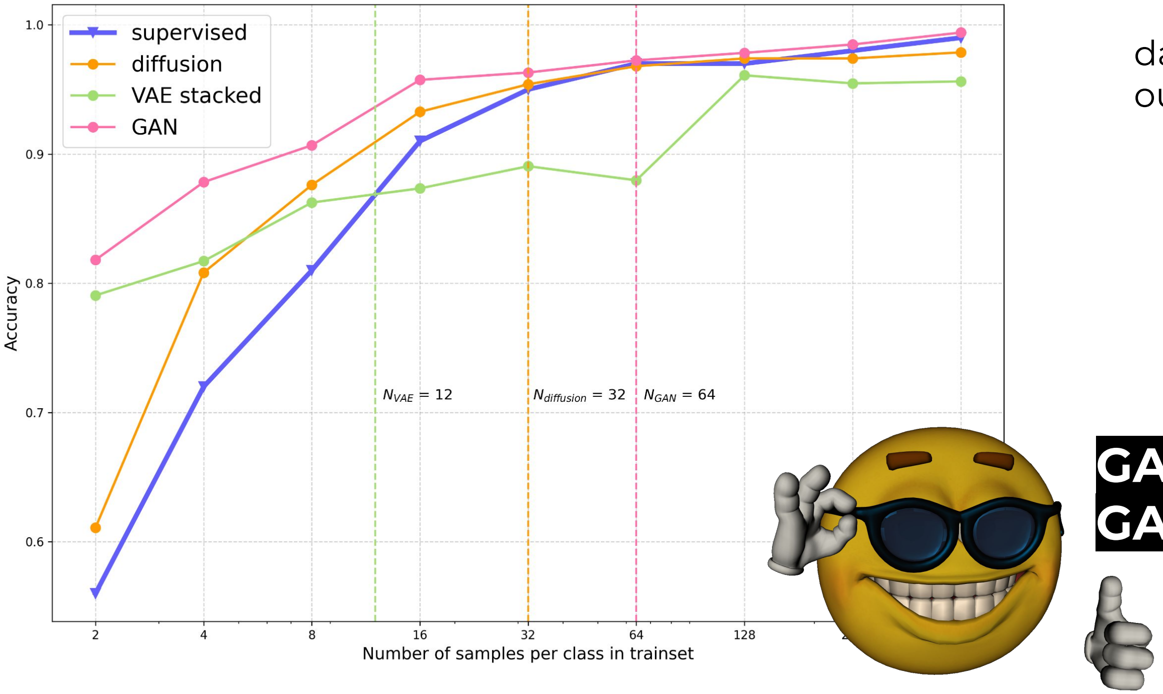
$$N_{\text{GAN}} \leqslant 64$$

$$N_{\text{diffusion}} \leqslant 32$$

$$N_{\text{VAE}} \leqslant 12$$

Training Results for MNIST

Training a **nonlinear** (Linear+ReLU+Linear) model on features extracted from generative models



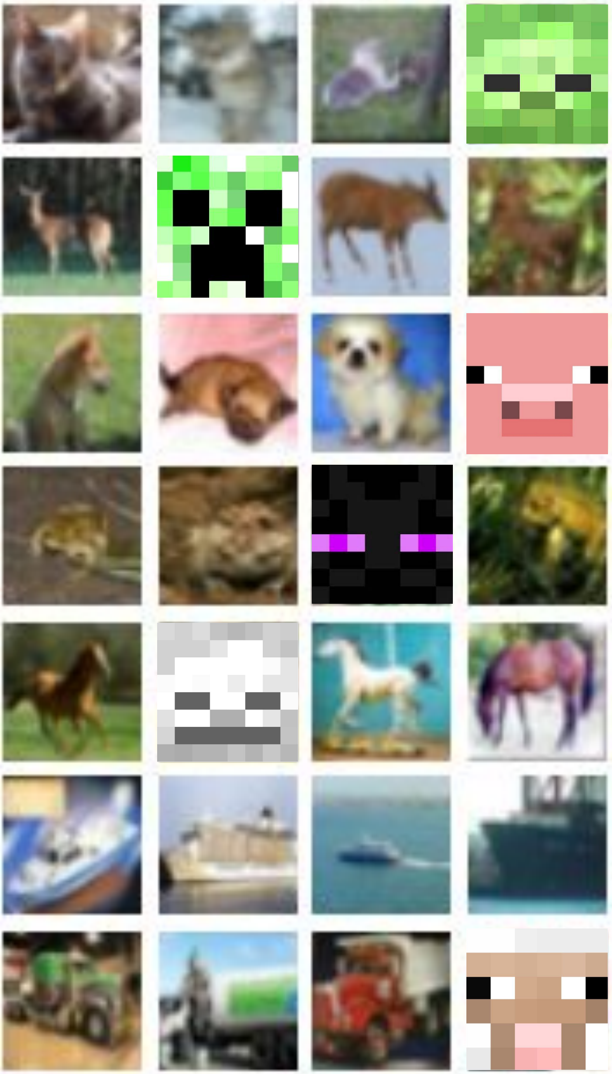
dataset sizes at which our models outperform ResNet-18:

$$N_{\mathrm{GAN}} \leqslant 64$$

$$N_{\mathrm{diffusion}} \leqslant 32$$

$$N_{\mathrm{VAE}} \leqslant 12$$

GAN is true GANgster



Results

CIFAR-10

Diffusion, VAE, GAN models results and comparison of their performance with resnet-18

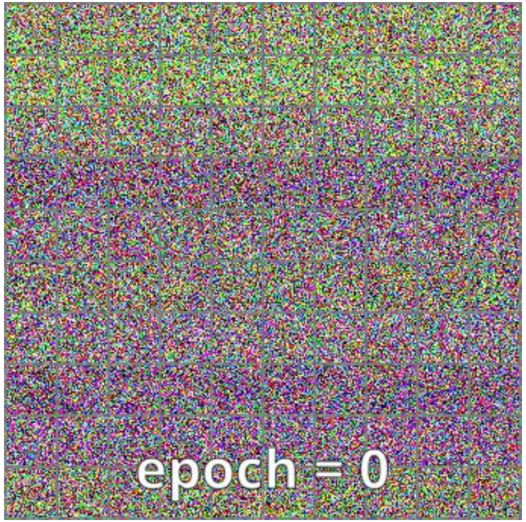
Generation quality

Visual estimation of generation quality

CIFAR-10 results

diffusion

time = 2h 30 min

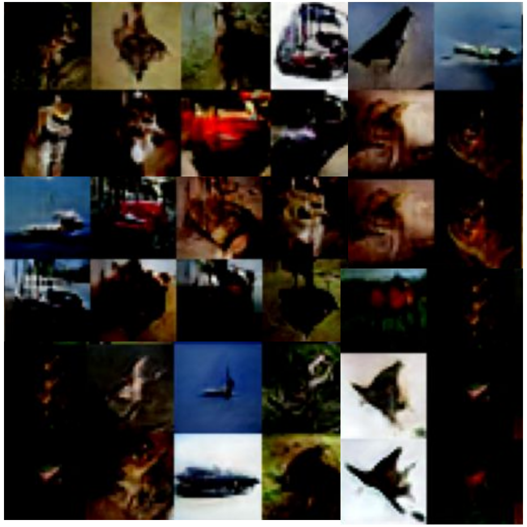


VAE

time = 10 min

GAN

time = 3 h, **A100**!

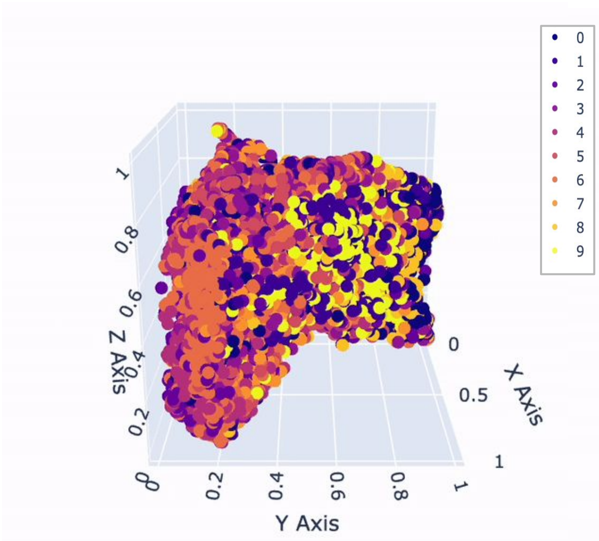


UMAP projection

Projection of features after training on **unlabeled** dataset, **CIFAR-10 dataset**

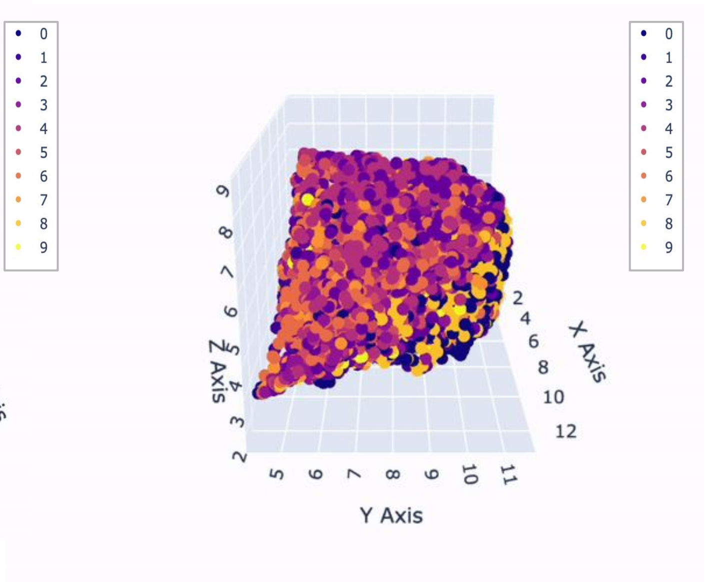
diffusion

feature dim = 512



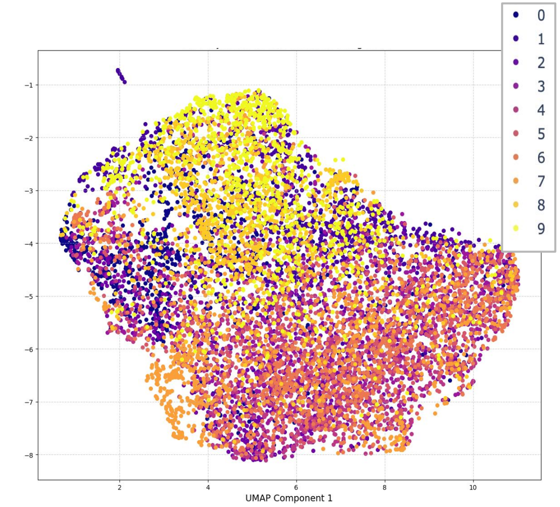
VQ-VAE *(encoder)*

feature dim = 1020



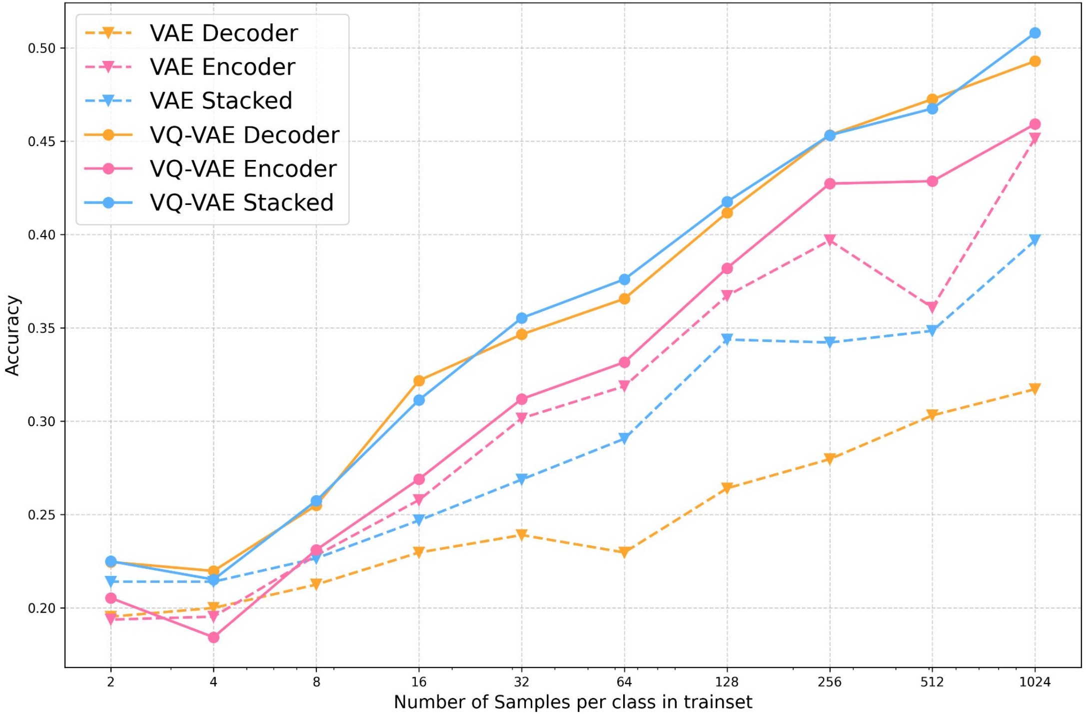
GAN

feature dim = 128



Training Results for CIFAR-100

CIFAR-10 results



VAE performed poorly on the CIFAR-10 dataset, so we tried the stronger **VQ-VAE model**

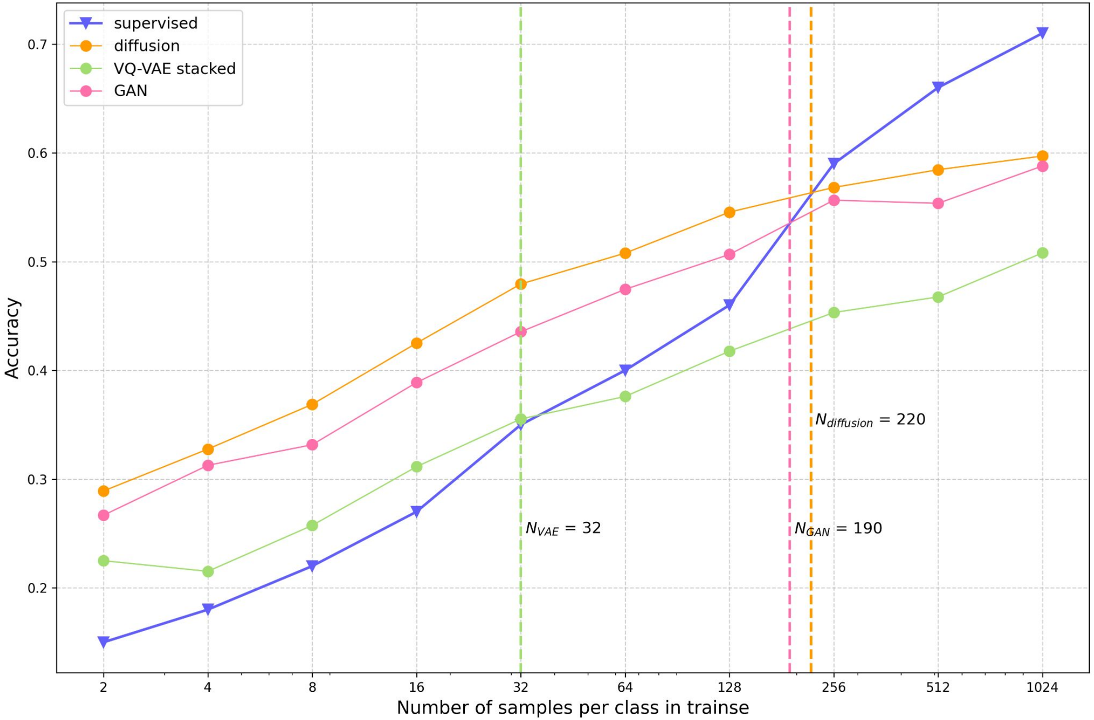
dataset size = 16

26% → **33%**

VAE VQ-VAE

Training Results for CIFAR-10

Training a **nonlinear** (Linear+ReLU+Linear) model on features extracted from generative models

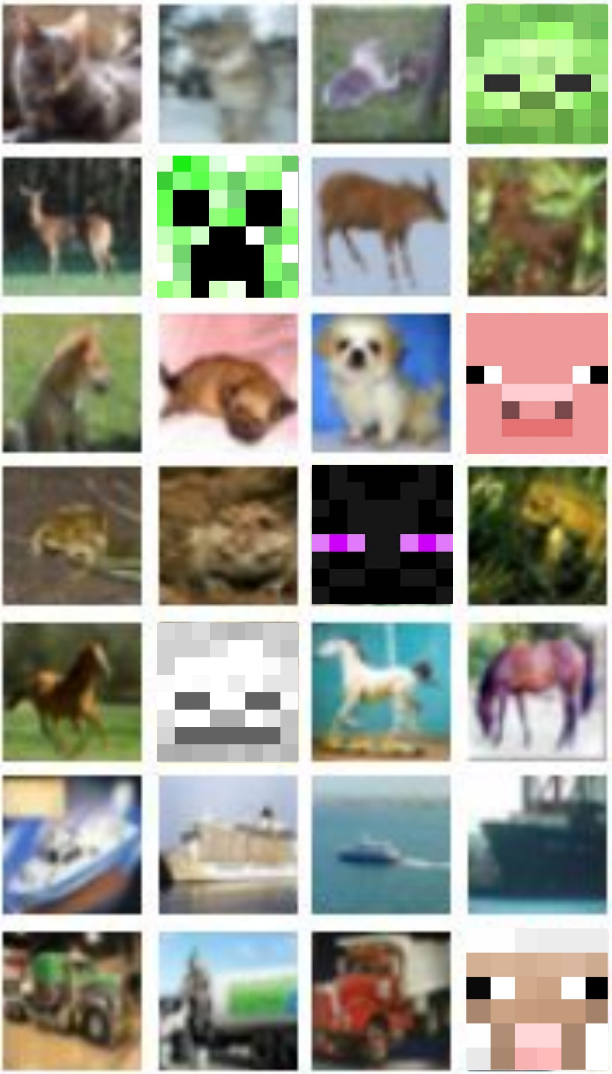


dataset sizes at which the model outperforms ResNet-18

$N_{\text{diffusion}} \leqslant 220$

$N_{\text{GAN}} \leqslant 190$

$N_{\text{VQ-VAE}} \leqslant 32$



Results

CIFAR-100

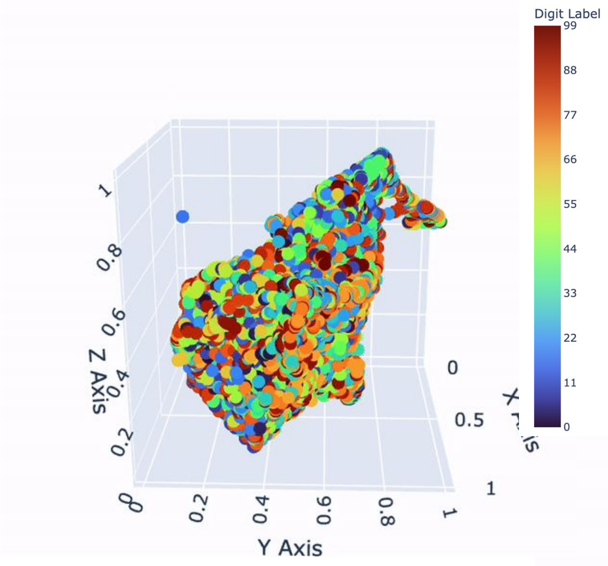
Diffusion, VAE, GAN models results and comparison of their performance with resnet-18

UMAP projection

Projection of features after training on **unlabeled** dataset, **CIFAR-100 dataset**

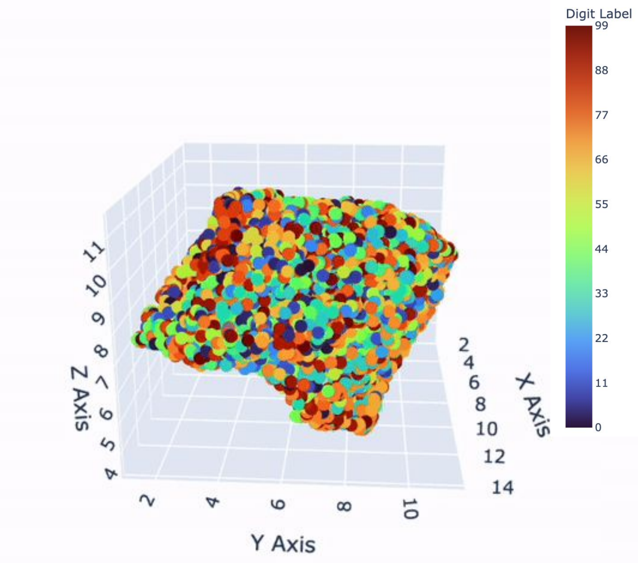
diffusion

feature dim = 512



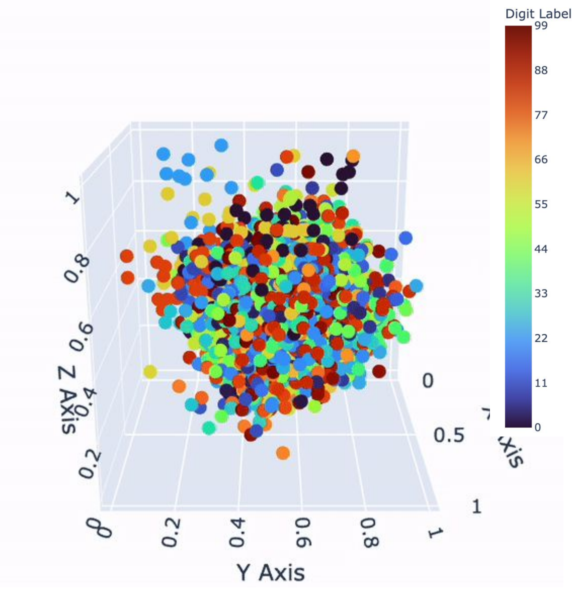
VQ-VAE *(stacked)*

feature dim = 1020



GAN

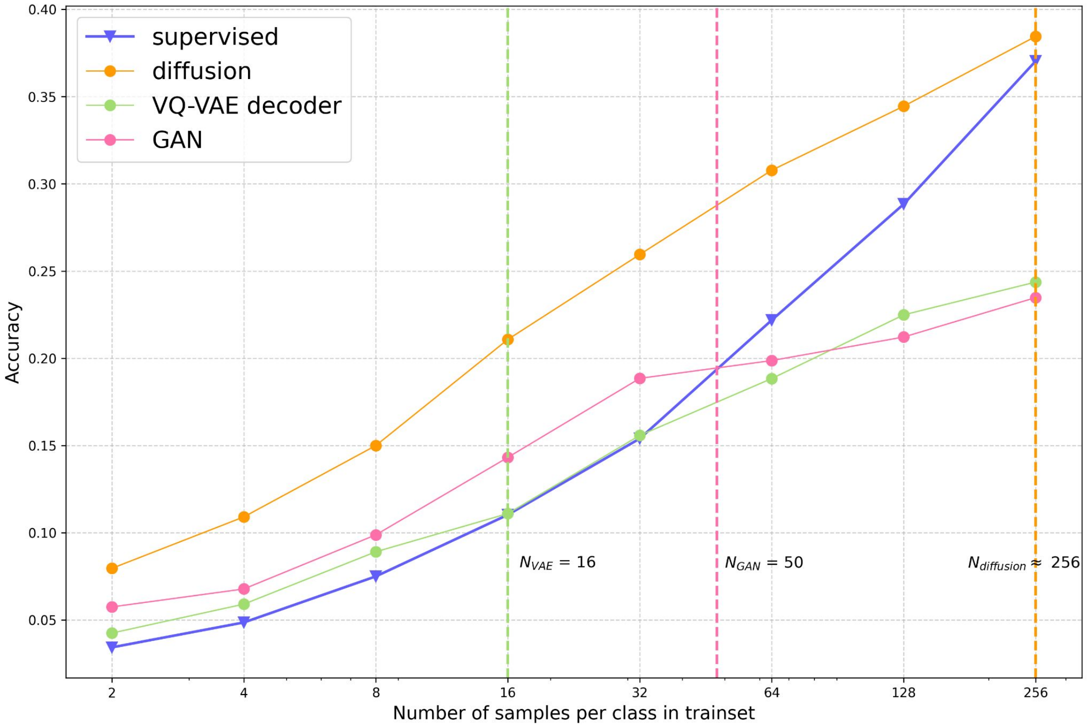
feature dim = 128



Training Results for CIFAR-100

CIFAR-100 results

Training a **nonlinear** (Linear+ReLU+Linear) model on features extracted from generative models

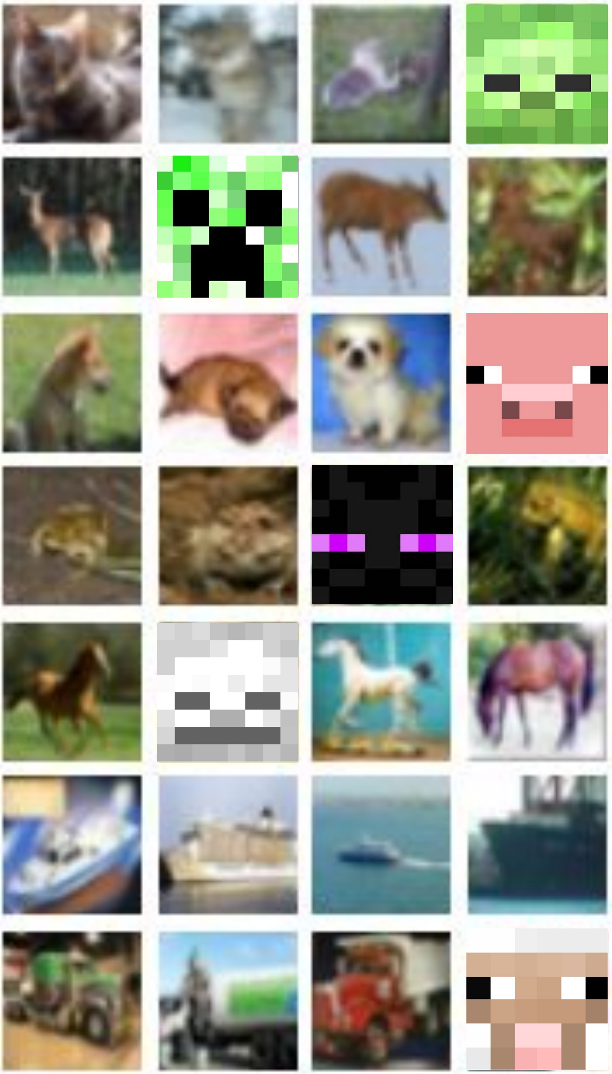


dataset sizes at which the model outperforms ResNet-18

$N_{\text{diffusion}} \lesssim 256$

$N_{\text{GAN}} \leqslant 50$

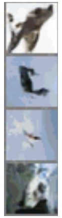
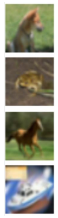
$N_{\text{VQ-VAE}} \leqslant 16$



Conclusion

Main results of our research

Main Results

		trainset size per class	**ResNet-18** *baseline*	**diffusion**	**VQ-VAE/VAE**	**GAN**
	**MNIST**	2	56%	32%	80%	**82%**
		4	72%	36%	83%	**88%**
		8	81%	36%	86%	**90%**
	**CIFAR-10**	2	15%	**29%**	22%	27%
		4	18%	**33%**	22%	31%
		128	46%	**54%**	42%	51%
	**CIFAR-100**	2	3%	**8%**	4%	6%
		8	8%	**15%**	9%	10%
		16	11%	**21%**	11%	14%

Questions?

Artem Alekseev
Artem.Alekseev@skoltech.ru
Data Science

Irina Lebedeva
Irina.Lebedeva@skoltech.ru
IoT and Wireless Technologies

Ignat Melnikov
Ignat.Melnikov@skoltech.ru
Data Science

Viktoriia Zinkovich
Viktoriia.Zinkovich@skoltech.ru
Data Science

Kamil Garifullin
Kamil.Garifullin@skoltech.ru
Data Science

Generative models in papers

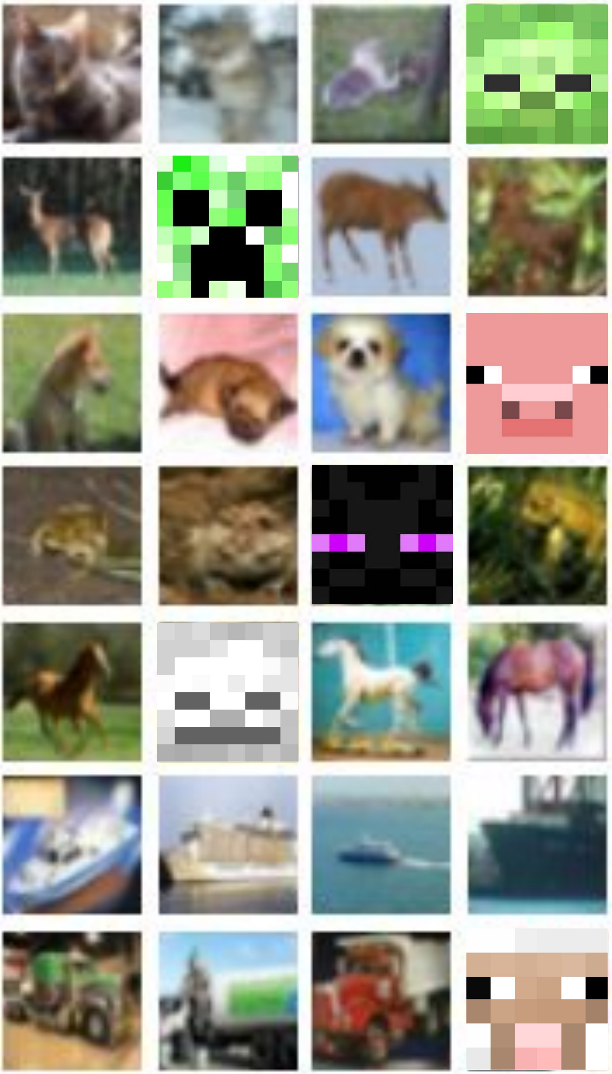
Our generative models

References:

[1] **The Unreasonable Effectiveness of Deep Features as a Perceptual Metric.** R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang. 2018.

[2] **Structured Denoising Diffusion Models in Discrete State-Spaces.** J. Austin, D.D. Johnson, J. Ho, D. Tarlow, R. van den Berg. 2023.

[3] **DatasetGAN: Efficient labeled data factory with minimal human effort**, Zhang, Y., Ling, H., Gao, J., Yin, K., Lafleche, J. F., Barriuso, A., Fidler, S. 2021.



Appendix

Forward diffusion process

Evolution of the project over two courses: ML and DL

ML 2024 Course

➔ **MNIST**:
- ✅ VAE
- ❌ VQ-VAE
- ❌ Diffusion
- ❌ GAN

➔ **CIFAR10**:
- ✅ VAE
- ✅ Diffusion
- ❌ VQ-VAE
- ❌ GAN

➔ **CIFAR100**:
- ❌ VAE
- ❌ VQ-VAE
- ❌ Diffusion
- ❌ GAN

DL 2024 Course

Evolution of the project over two courses: ML and DL

ML 2024 Course

→ **MNIST**:
- ✅ VAE
- ❌ VQ-VAE
- ~ Diffusion
- ❌ GAN

→ **CIFAR10**:
- ✅ VAE
- ✅ Diffusion
- ❌ VQ-VAE
- ❌ GAN

→ **CIFAR100**:
- ❌ VQ-VAE
- ❌ Diffusion
- ❌ GAN

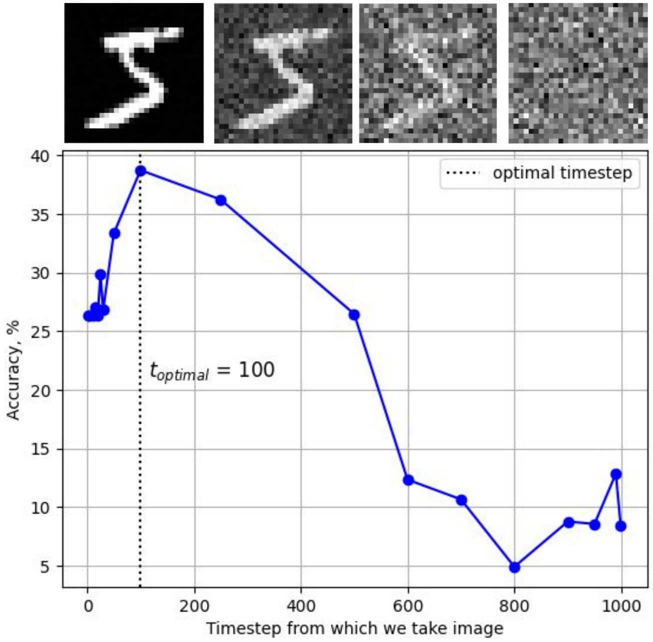
DL 2024 Course

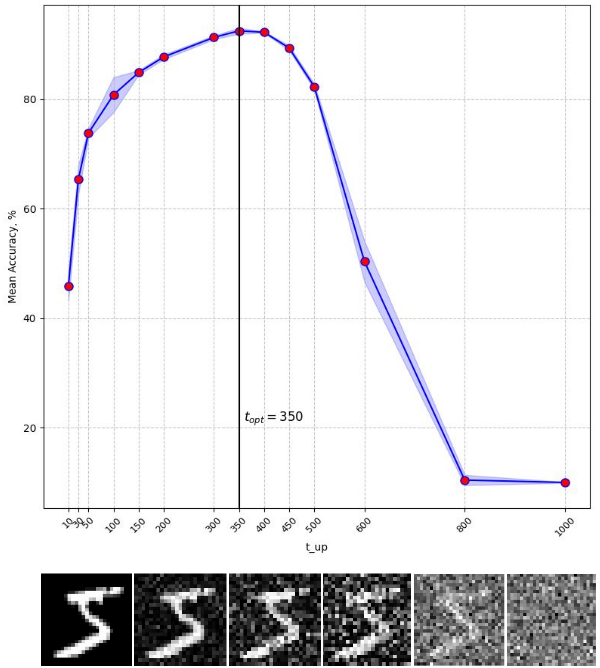
→ **MNIST**:
- ✅ VAE
- ✅ VQ-VAE
- ✅ Diffusion
- ✅ GAN

→ **CIFAR10**:
- ✅ VAE
- ✅ Diffusion
- ✅ VQ-VAE
- ✅ GAN

→ **CIFAR100**:
- ✅ VQ-VAE
- ✅ Diffusion
- ✅ GAN

MNIST: $t_{opt} = 350$

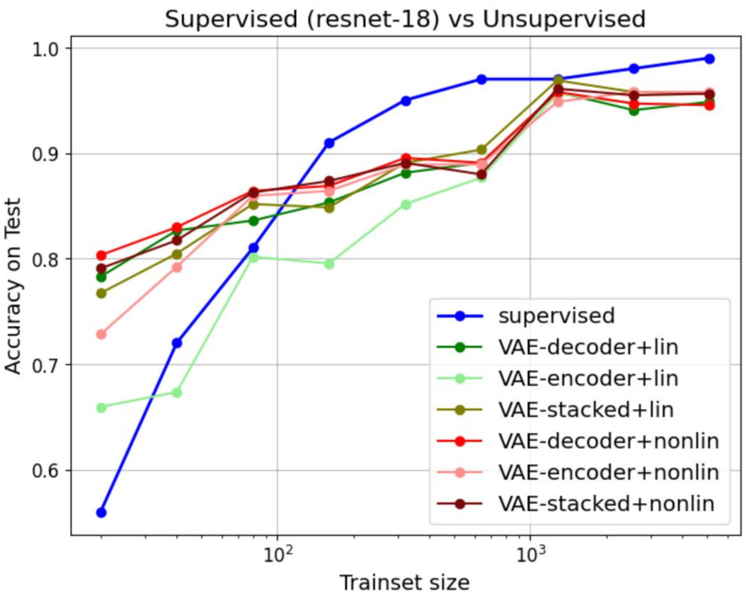




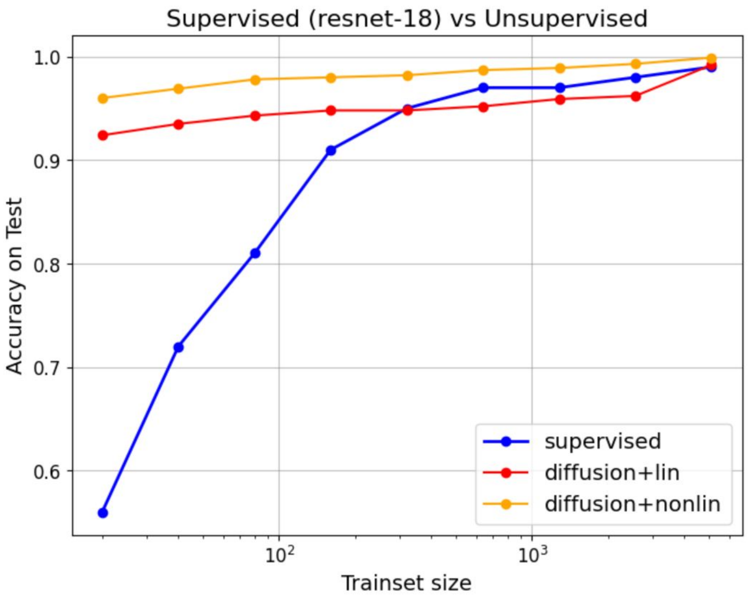
Training Results

Training **linear** (Linear) and **nonlinear** (Linear+ReLU+Linear) models on features from generative models

VAE



Diffusion



Main Results

With extremely **small dataset sizes**, our method **bypasses resnet-18** (our benchmark), which reinforces the initial hypothesis

MNIST 

trainset size per class	resnet-18	VAE	diffusion	GAN
2	56%	**80%**	**61%**	**82%**
4	72%	**83%**	**81%**	**88%**
8	81%	**86%**	**88%**	**90%**

CIFAR-10 

trainset size per class	resnet-18	VQ-VAE	diffusion	GAN
2	15%	**22%**	**29%**	**27%**
4	18%	**22%**	**33%**	**31%**
...	...	...	...	...
128	46%	**42%**	**54%**	**51%**