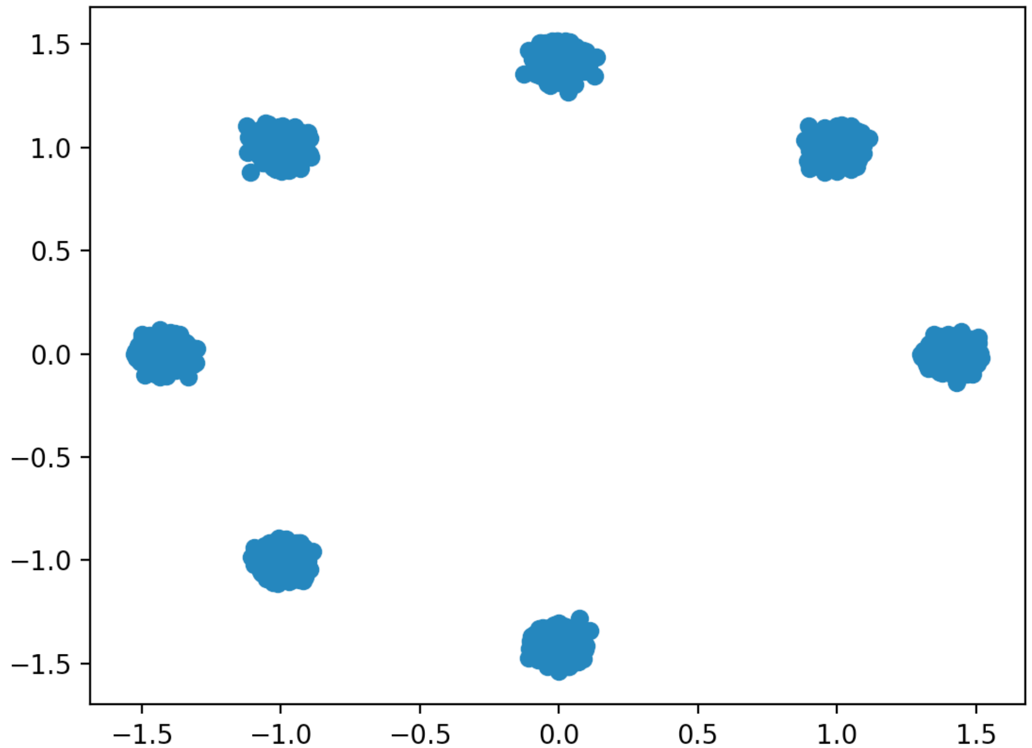
CREATIVITY IN GENERATIVE ADVERSARIAL NETWORKS

Güneş Yurdakul
Advisor: Liu Yuejiang
Prof: Alexandre Alahi

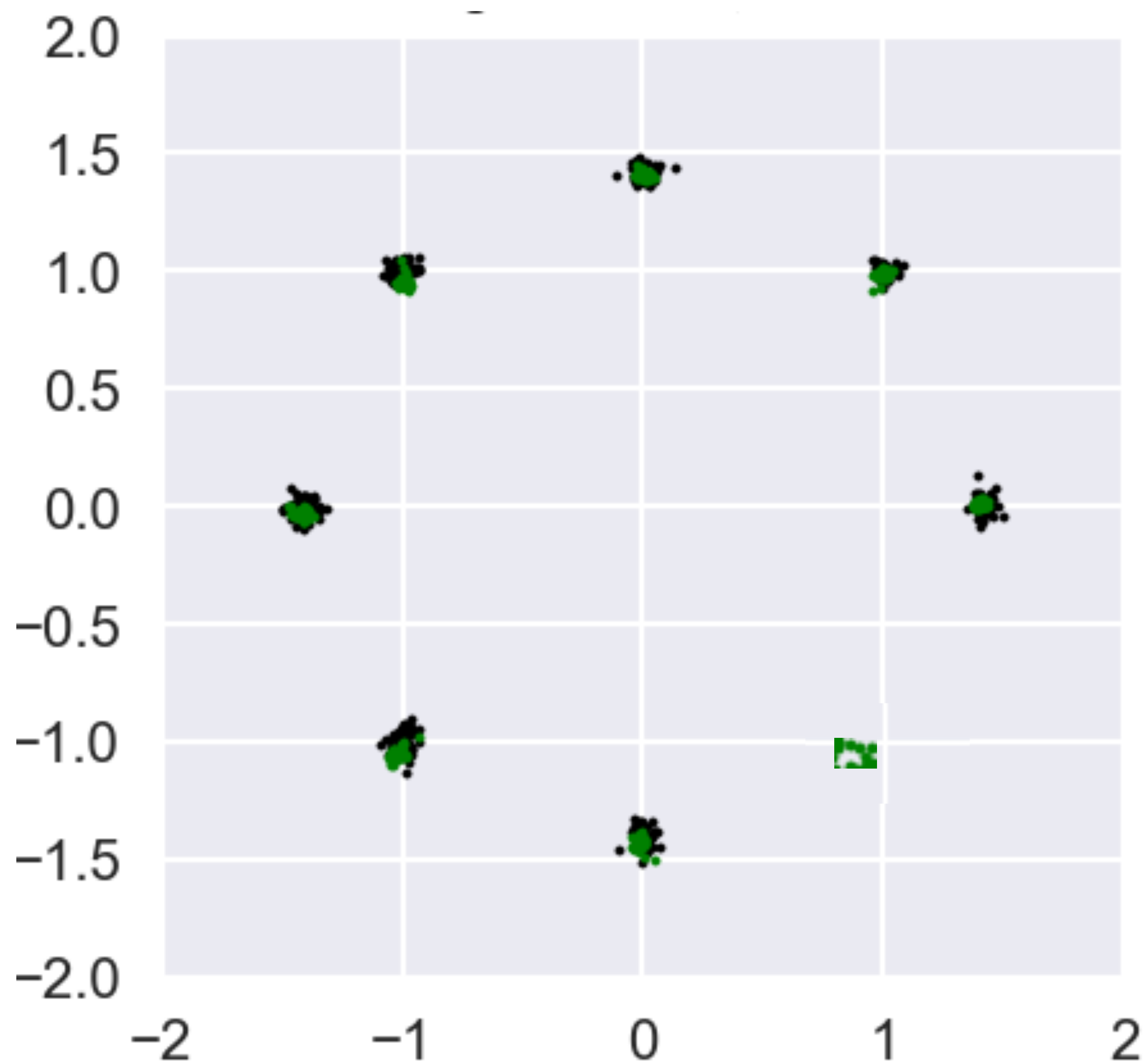
PROBLEM

- Explore creativity in generative models (GAN)
- Working on finding a way to generate samples from unknown classes
- VAEs and GANs require many samples from each class to generate samples from these classes.
- Experimenting on one-shot and few-shot settings.

INPUT



WHAT WE WANT



DesIGN: DESIGN INSPIRATION FROM GENERATIVE NETWORKS

CLASSIFICATION LOSS

$$\mathcal{L}_D = \lambda_{D_r}\mathcal{L}_{D\text{ real/fake}} + \lambda_{D_b}\mathcal{L}_{D\text{classif}} \quad \text{with}$$
$$\mathcal{L}_{D\text{ classif}} = -\sum_{x_i \in \mathcal{D}} \log(\operatorname{softmax}(D_{b,\hat{c}_i}(x_i))),$$

- STYLE GAN:

- Fashion
Attribute discovery dataset

- ACGAN

- shape and texture classification on the real images on top of predicting real/fake discrimination.

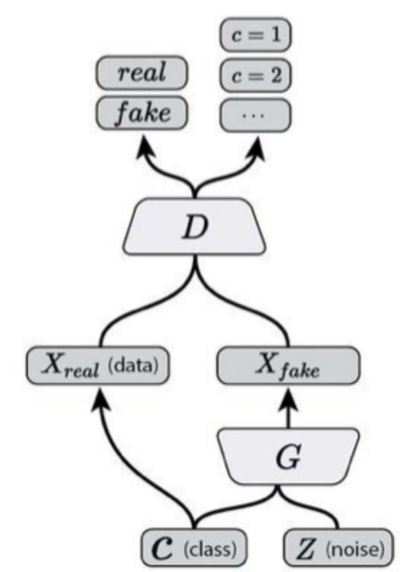
CREATIVITY LOSS

to make samples deviate from the training samples.

$$\mathcal{L}_G = \lambda_{G_r}\mathcal{L}_{G\text{ real/fake}} + \lambda_{G_e}\mathcal{L}_{G\text{ creativity}}$$

$$\mathcal{L}_{\text{MCE}} = -\sum_i \frac{1}{K}\log\operatorname{softmax}(D_b(x_i))$$
$$= -\sum_i \frac{1}{K}\log\left(\frac{e^{D_{b,\hat{c}_i}(x_i)}}{\sum_{k=1}^{K} e^{D_{b,k}(x_i)}}\right)$$

Auxiliary Classifier GAN
(Odena, et al., 2016)



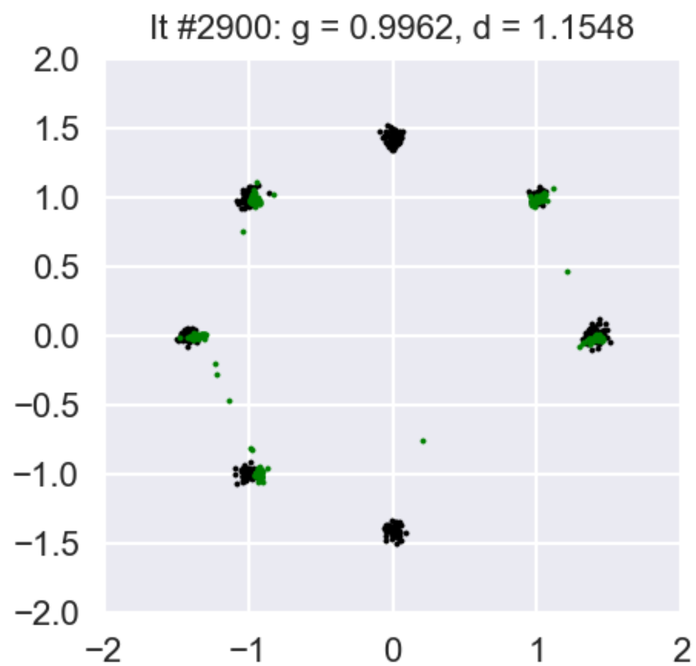
Sbai, O., Elhoseiny, M., Bordes, A., LeCun, Y., & Couprie, C. (**2018**). Design: Design inspiration from generative networks. In *Proceedings of the European Conference on Computer Vision (**ECCV**)* (pp. 0-0).

GENERATED SAMPLES

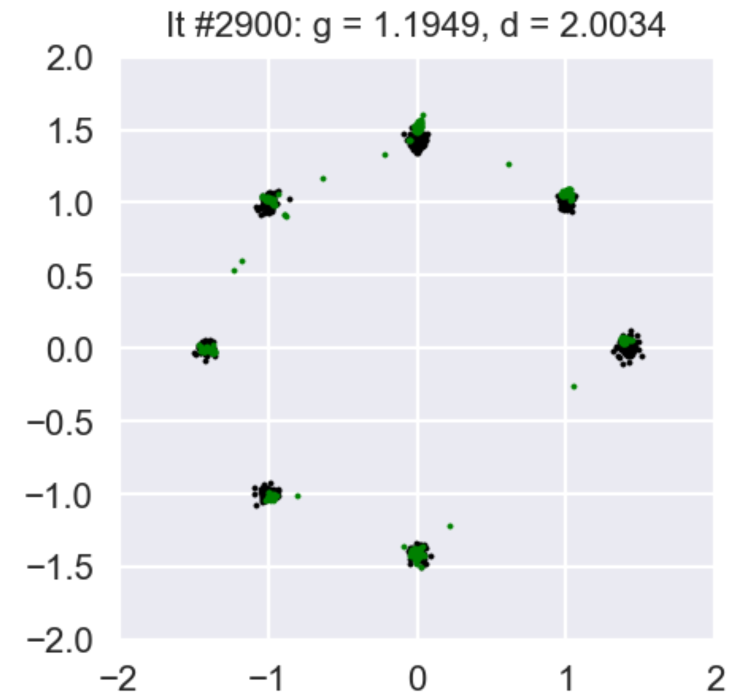
EVALUATION METRICS

- Mean nearest neighbors distance score
- Human evaluation (survey)
  - Q1:how do you like this design overall on a scale from 1 to 5?
  - Q2/Q3:rate the novelty of shape(Q2) and texture(Q3) from 1 to 5.
  - Q4/Q5: rate the complexity of shape (Q4) and texture (Q5) from 1 to 5.
  - Q6: Do you think this image was created by a fashion designer or generated by computer? (yes/no)

NORMAL GAN



STYLE GAN



NORMAL GAN

STYLE GAN

2 classes missing

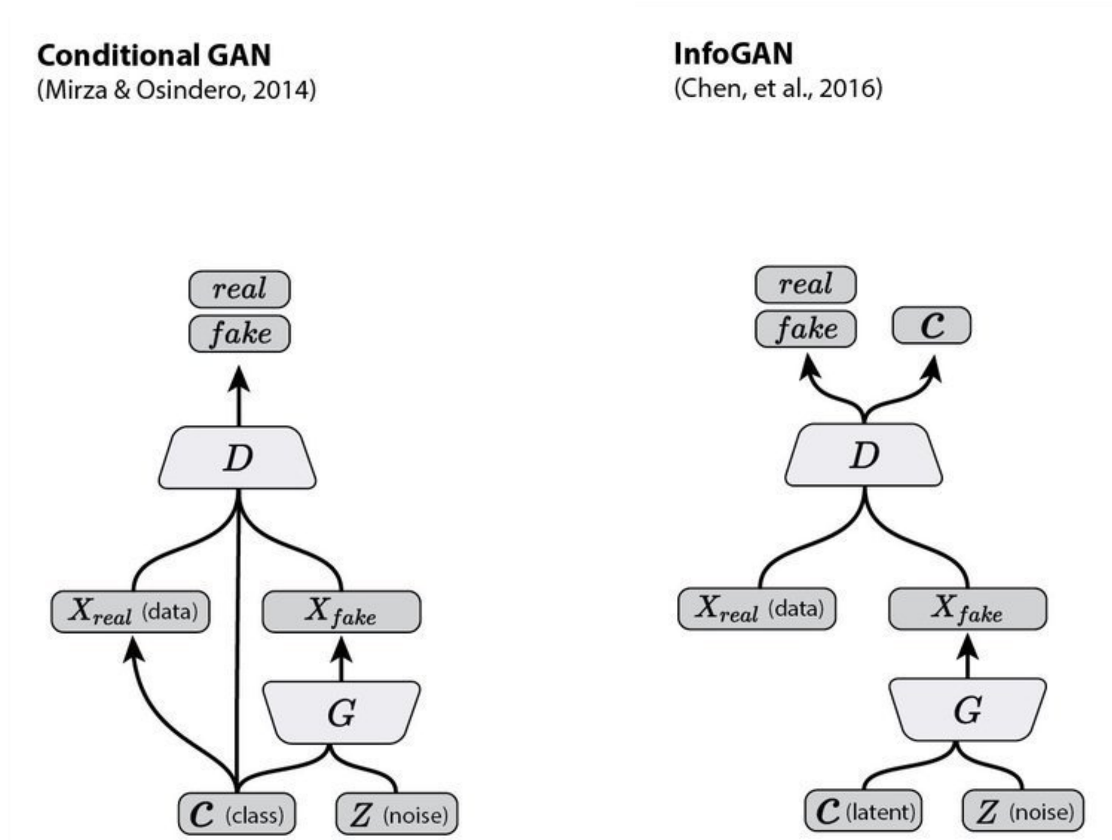
3 classes missing

WEAKNESS OF PREVIOUS WORK

- Evaluation methods
  - Hard to both define and evaluate creativity
  - Human study - subjective
- Creativity Loss
  - not effective

OUR METHOD

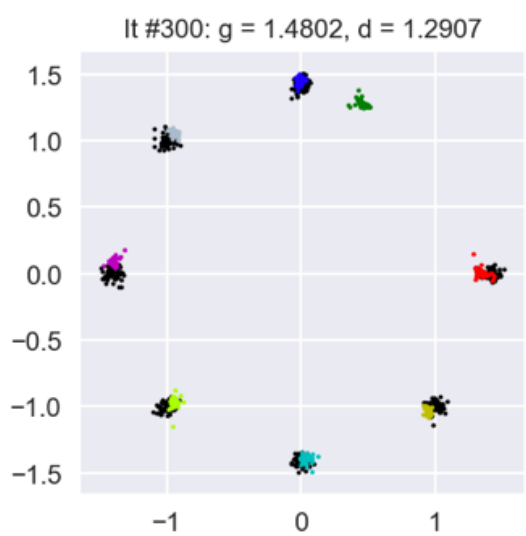
- We are working on finding a way to find semantic correlation between the latent (feature) space
- Experiments by removing real samples of a label:
  - CGAN
  - INFOGAN

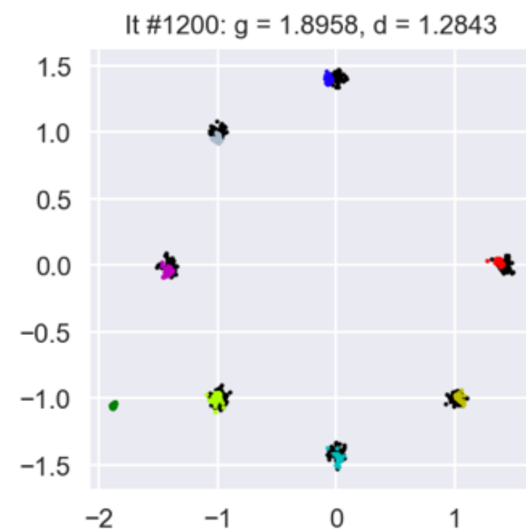


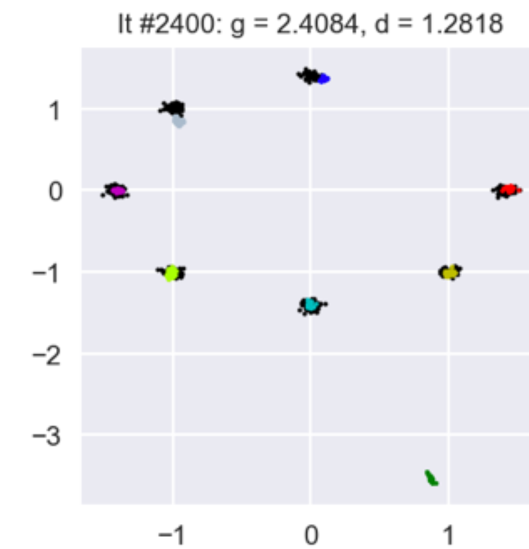
- Working on finding a way to generate samples from not introduced classes

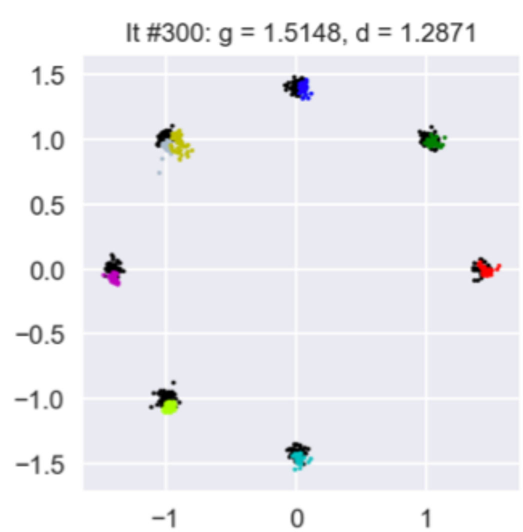
CONDITIONAL GAN

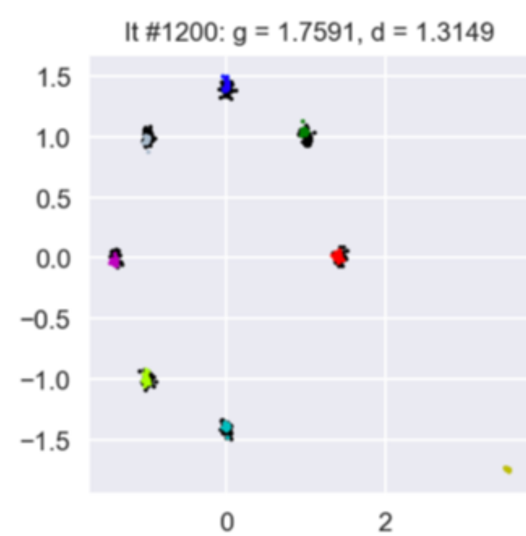
# of iterations: 300 | 1200 | 2400

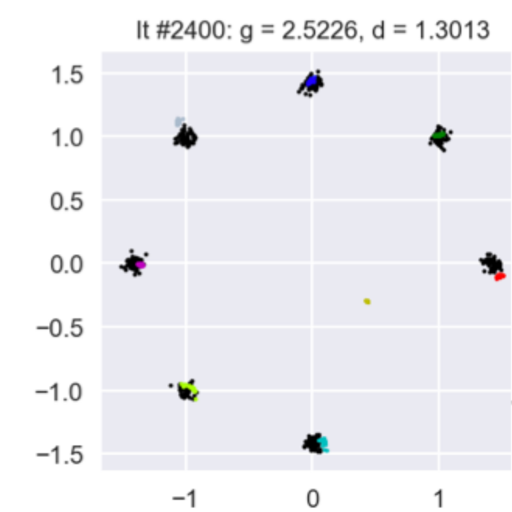




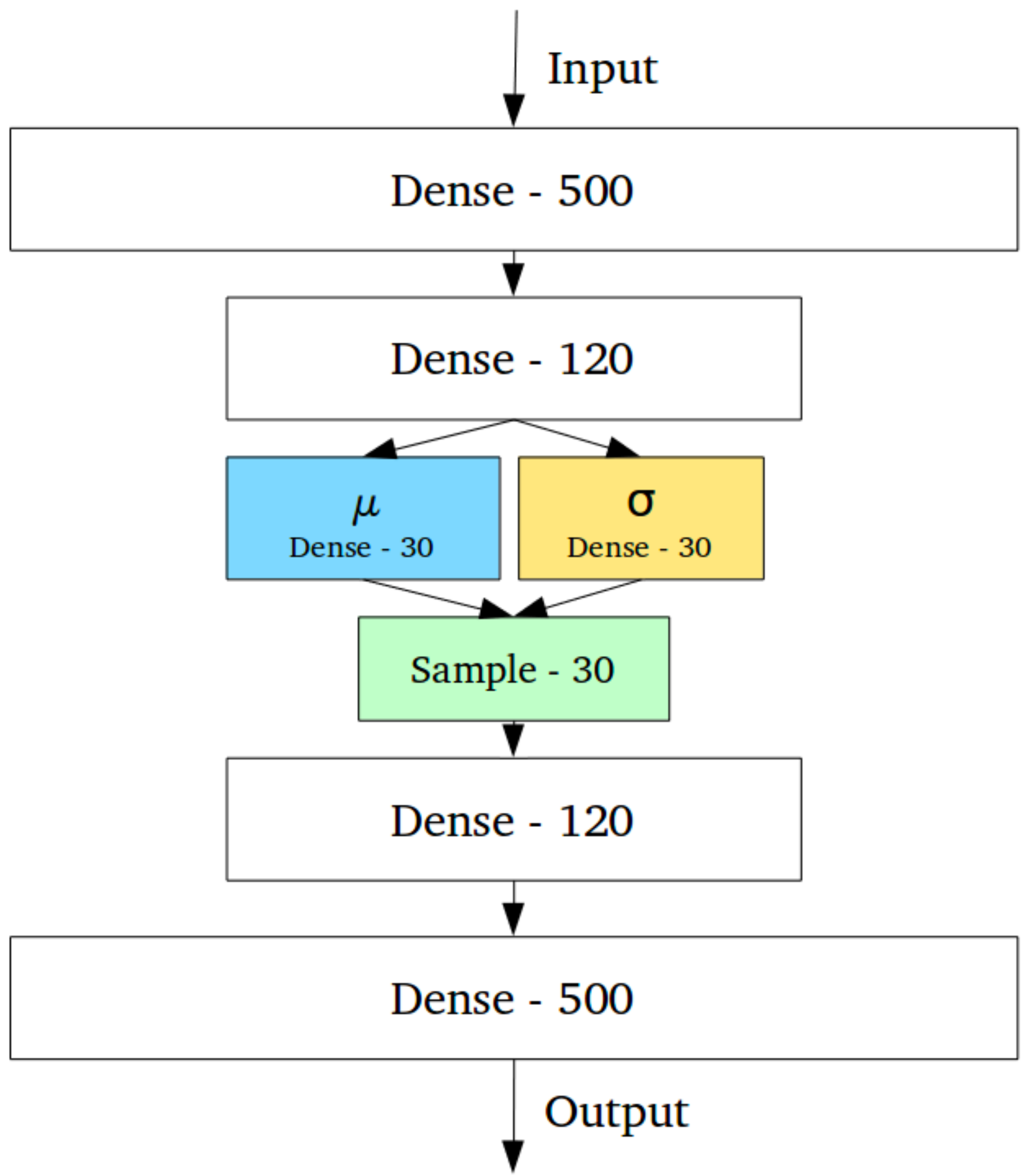








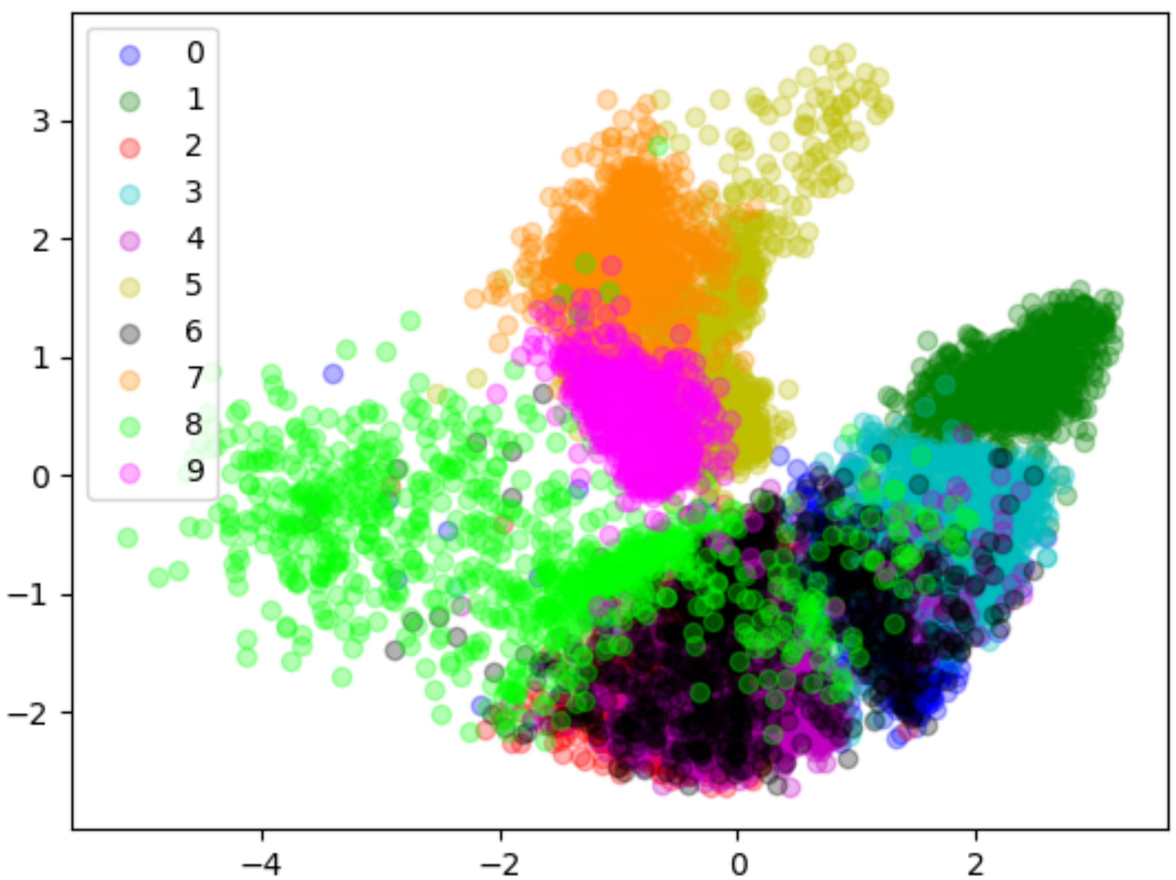
VAE

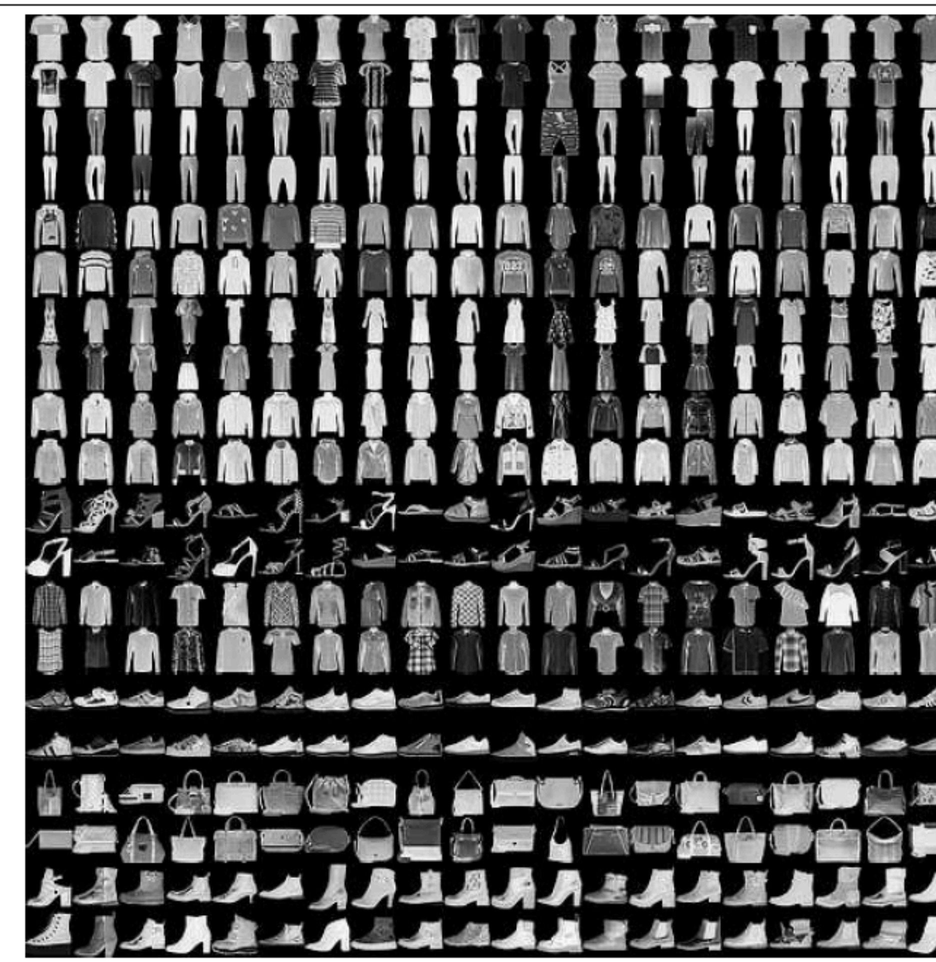


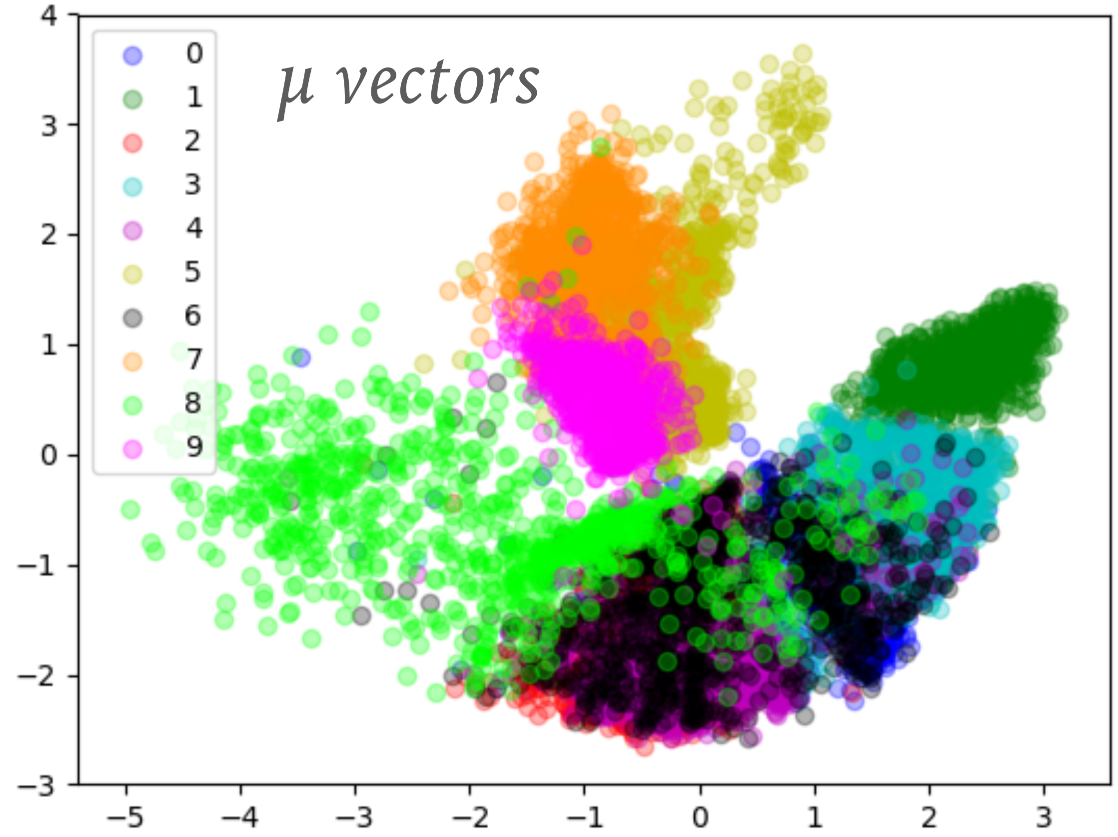
Input
Dense - 500
Dense - 120
μ
Dense - 30
σ
Dense - 30
Sample - 30
Dense - 120
Dense - 500
Output

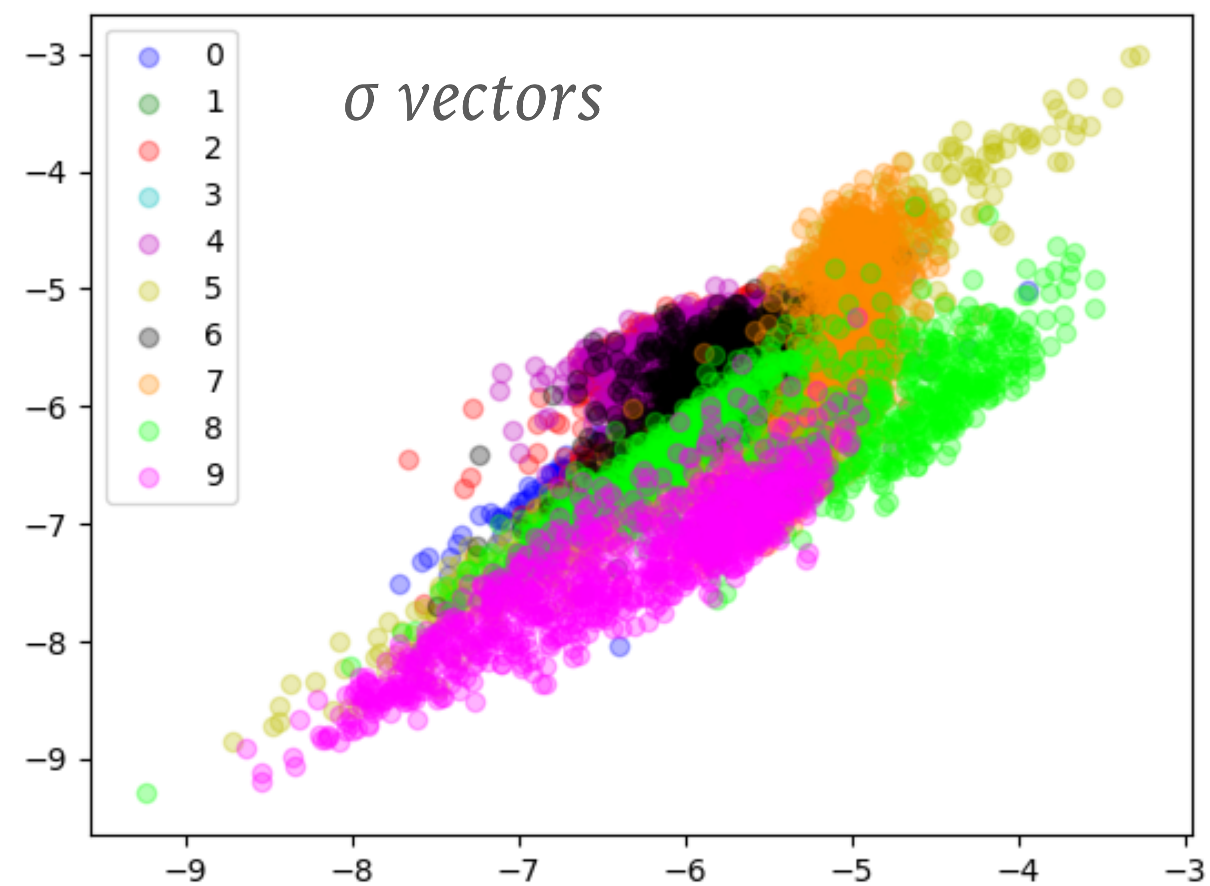
VAE

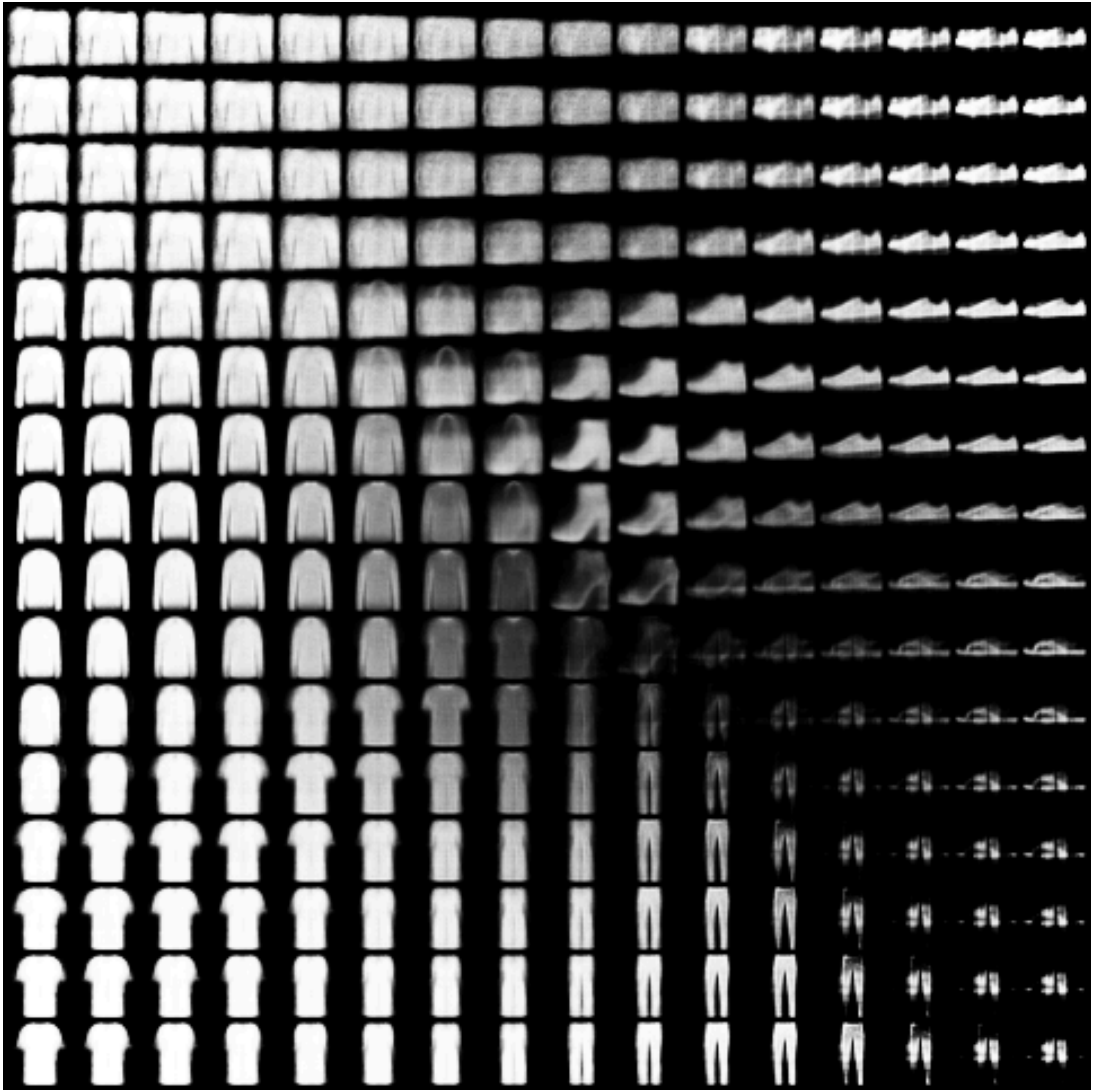
- Encoded vector z size:2
- Projecting z



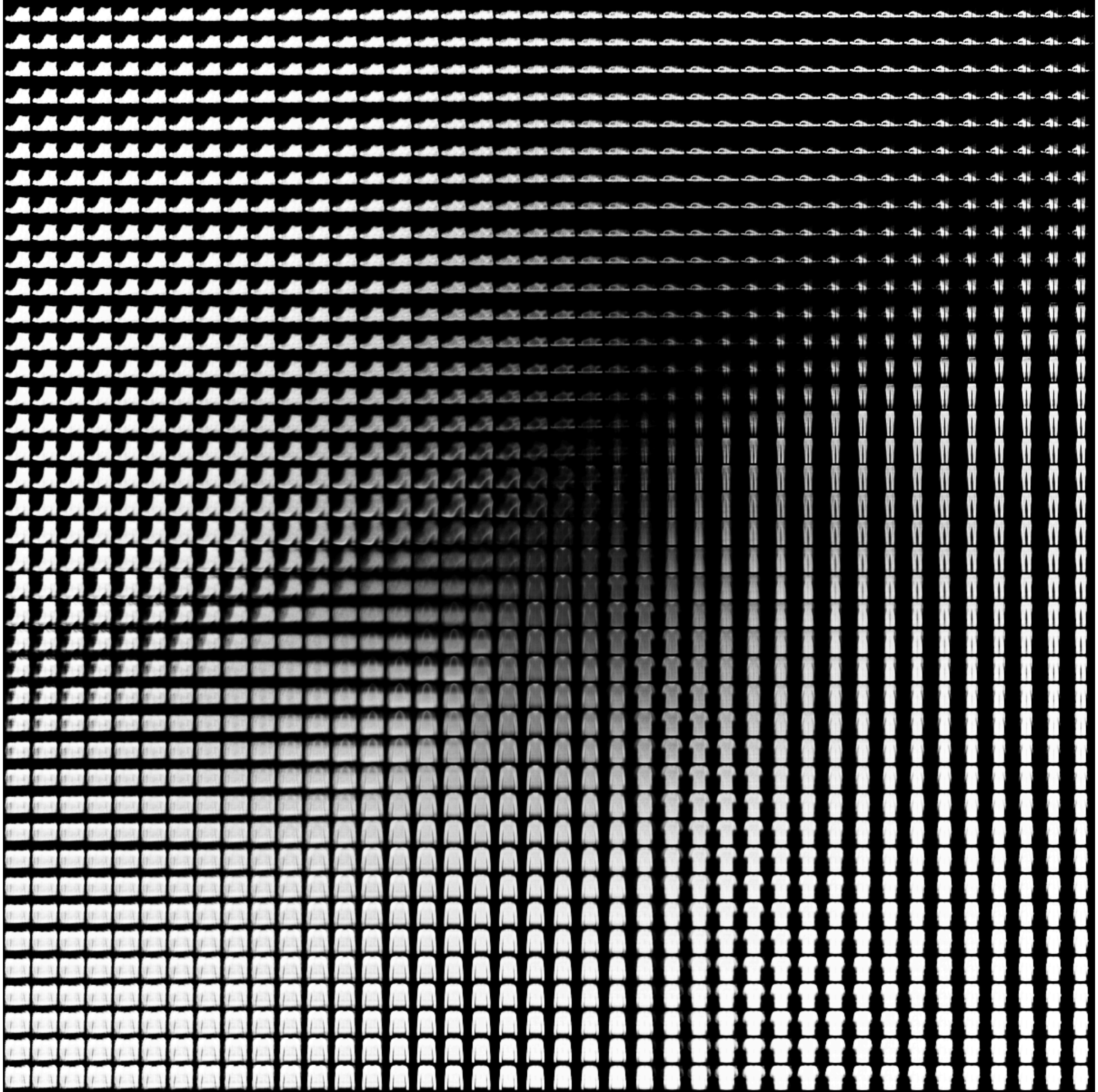
Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	


μ vectors
0
1
2
3
4
5
6
7
8
9

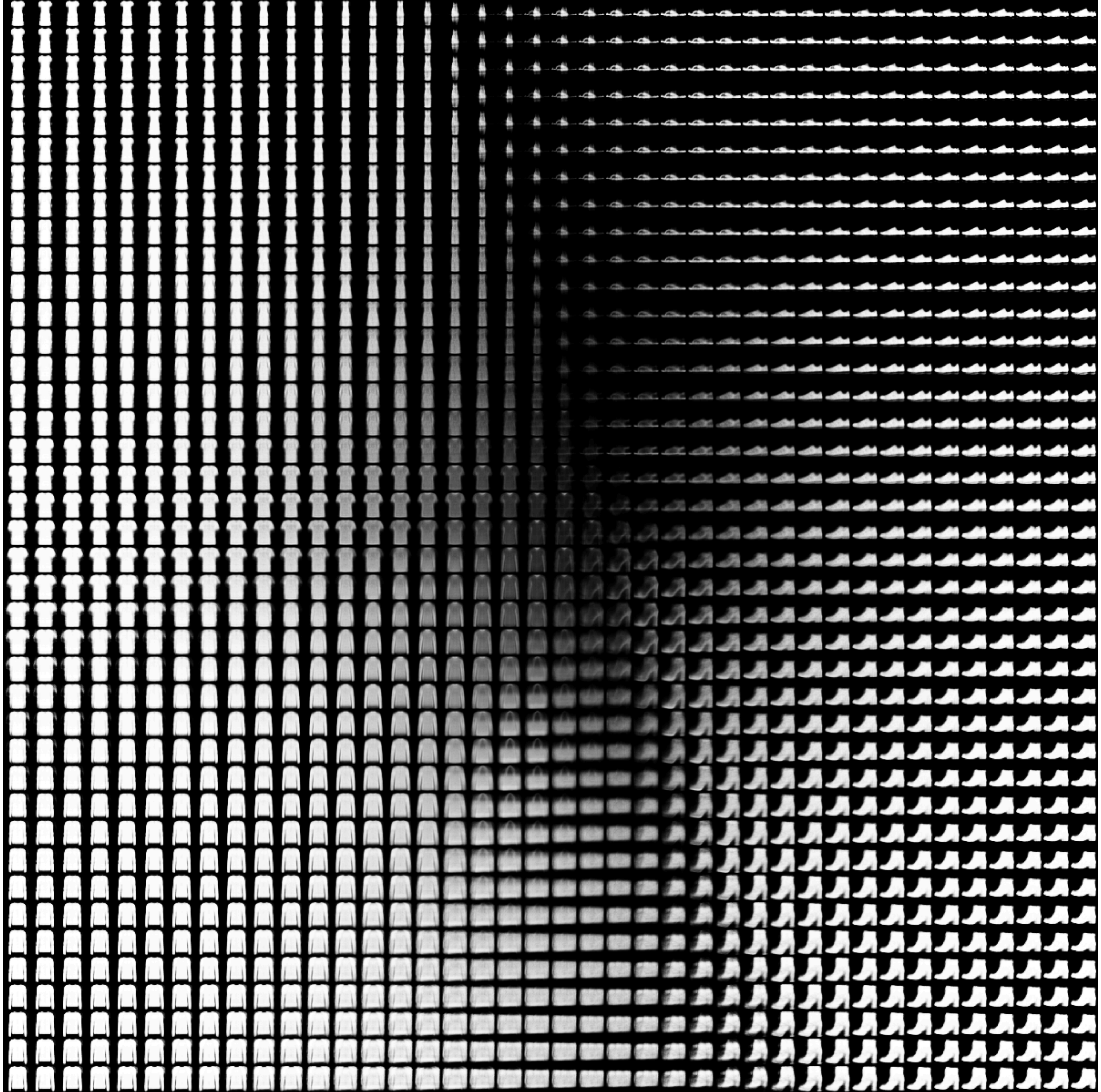

σ vectors
0
1
2
3
4
5
6
7
8
9



SNEAKARS MISSING IN TRAINING SET



PANTS MISSING IN TRAINING SET



ON FIRST-ORDER META-LEARNING ALGORITHMS

- Meta-Learning - “learning to learn”
- can adapt to new environments rapidly with a few training examples.

- MAML (Model-Agnostic Meta-Learning) [1]
- First-order MAML
- Reptile [2]

[1] Finn, C., Abbeel, P., & Levine, S. (2017, August). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1126-1135). JMLR. org.

[2] Nichol, A., Achiam, J., & Schulman, J. (2018). On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

REPTILE

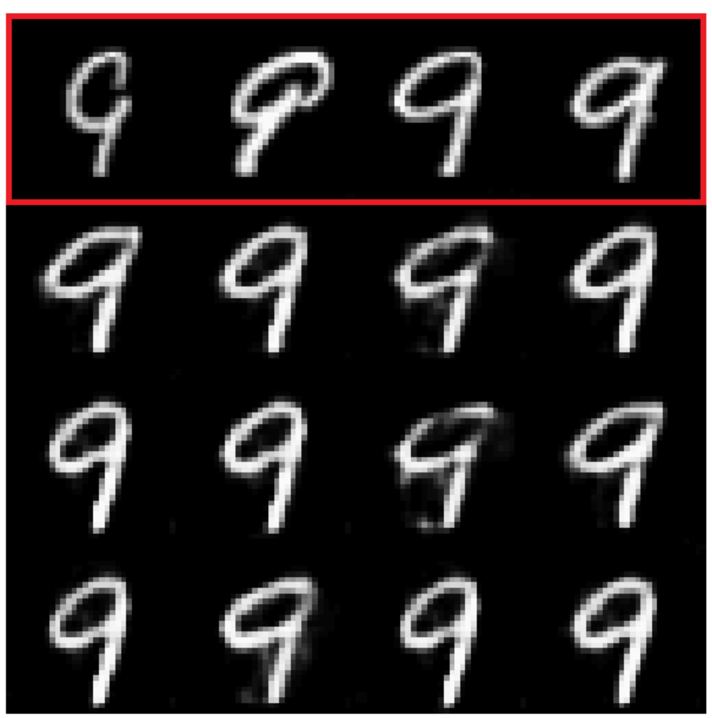
- repeatedly sampling a task
- training on it
- moving the initialization towards the trained weights on that task.
- it does not calculate any second derivatives.
- takes less computation and memory than MAML

Algorithm 1 Reptile (serial version)

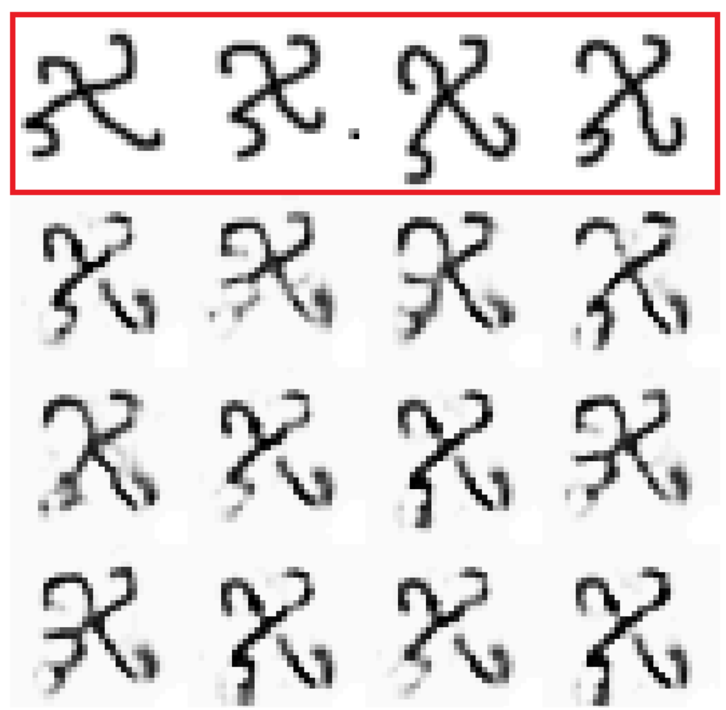
Initialize ϕ, the vector of initial parameters
for iteration $= 1, 2, \ldots$ **do**
    Sample task τ, corresponding to loss L_τ on weight vectors $\widetilde{\phi}$
    Compute $\widetilde{\phi} = U_\tau^k(\phi)$, denoting k steps of SGD or Adam
    Update $\phi \leftarrow \phi + \epsilon(\widetilde{\phi} - \phi)$
end for

FIGR: FEW-SHOT IMAGE GENERATION WITH REPTILE

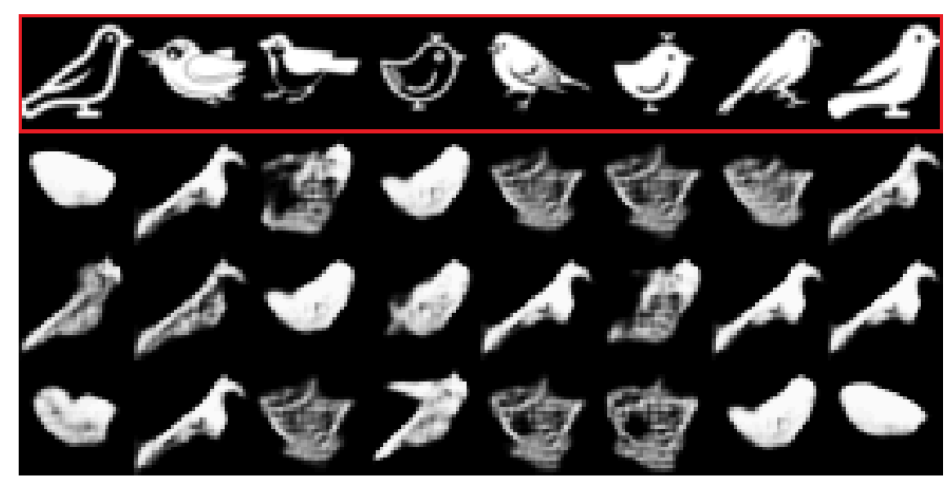
- a GAN meta-trained with Reptile.
- model generates novel images on both MNIST and Omniglot with 4 images from an unseen class.
- fine tuned on the data points in red circle.



MNIST; 50,000 update; 10 gradient steps



Omniglot; 230,000 update; 10 gradient steps



FIGR-8; 90,000 update; 10 gradient steps; $n = 8$

Clouâtre, L., & Demers, M. (2019). FIGR: Few-shot Image Generation with Reptile. *arXiv preprint arXiv:1901.02199*.

PROBLEMS

Algorithm 1: FIGR training

Initialize Φ_d, the discriminator parameter vector
Initialize Φ_g, the generator parameter vector
for iteration 1, 2, 3 ... **do**
  Make a copy of Φ_d resulting in W_d
  Make a copy of Φ_g resulting in W_g
  Sample task τ
  Sample n images from X_τ resulting x_τ
  for $K > 1$ iterations **do**
    Generate latent vector z
    Generate fake images y with z and W_g
    Perform step of SGD update on W_d with Wasserstein GP loss and x_τ and y
    Generate latent vector z
    Perform step of SGD update on W_g with Wasserstein loss and z
  end for
  Set Φ_d gradient to be Φ_d - W_d
  Perform step of Adam update on Φ_d
  Set Φ_g gradient to be Φ_g - W_g
  Perform step of Adam update on Φ_g
end for

Algorithm 2: FIGR generation

Using W_d, a copy of the meta-trained Φ_d
Using W_g, a copy of the meta-trained Φ_g
Sample test task τ
Sample n images as x_τ from X_τ
for $K >= 1$ iterations **do**
  Generate latent vector z
  Generate fake images y with z and W_g
  Perform step of SGD update on W_d with Wasserstein GP loss and x_τ and y
  Generate latent vector z
  Perform step of SGD update on W_g with Wasserstein loss and z
end for
Generate latent vector z
Generate fake images y

PROBLEMS WITH THE MODEL

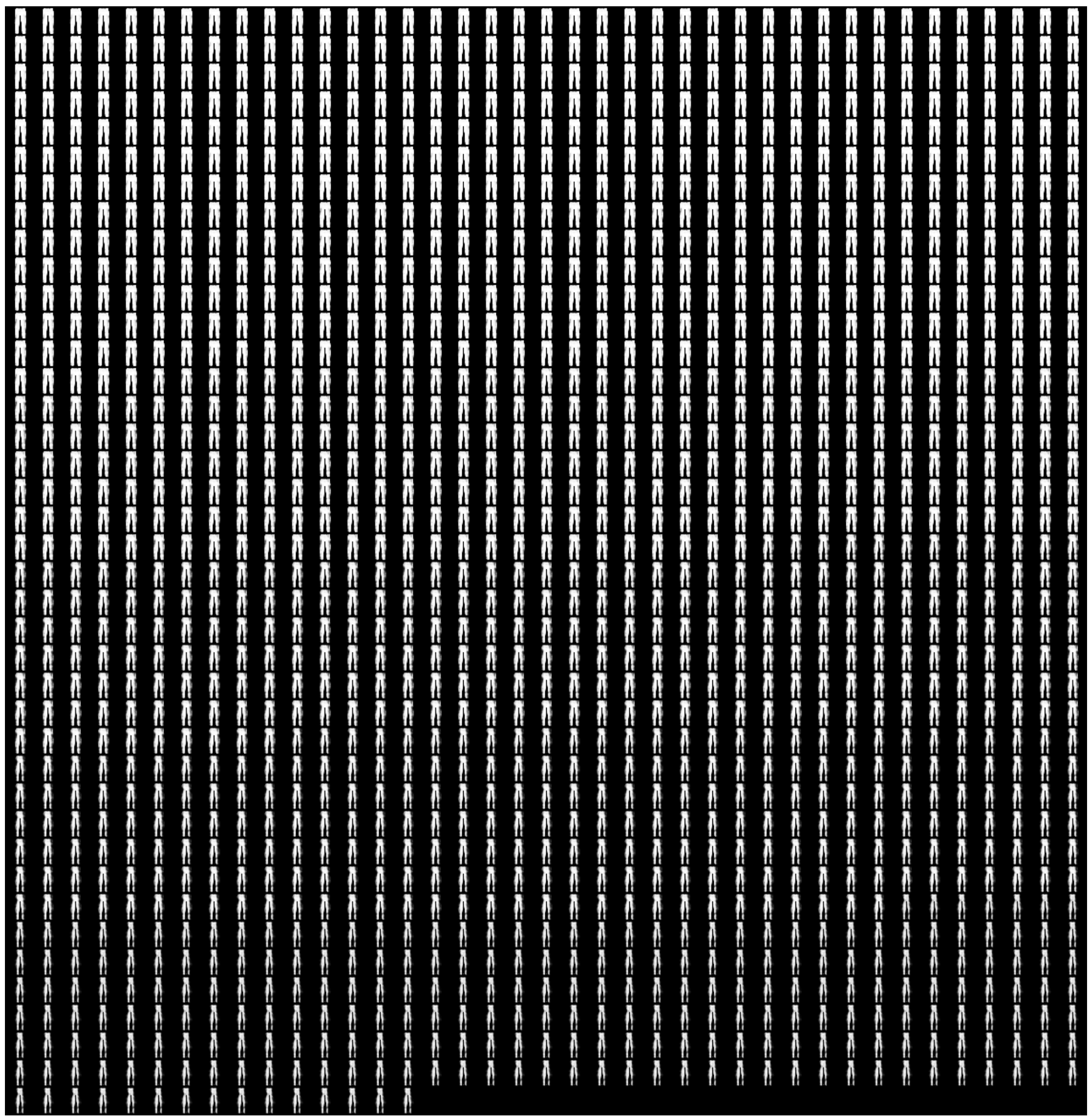
- Training a model - 250, 000 meta-training steps with n = 4 on Omniglot takes **125 hours ! (Tesla V100 on Google Cloud Platform (GCP))**
  - Wasserstein Loss with Gradient Penalty
  - Residual neural networks
    - Both generator and discriminator
- Human Evaluation
  - %50 of images (Omniglot) can fool human judge
  - None of the images (FIGR-8) generated can fool human judges.
  - No evaluation on novelty or creativity of created images.

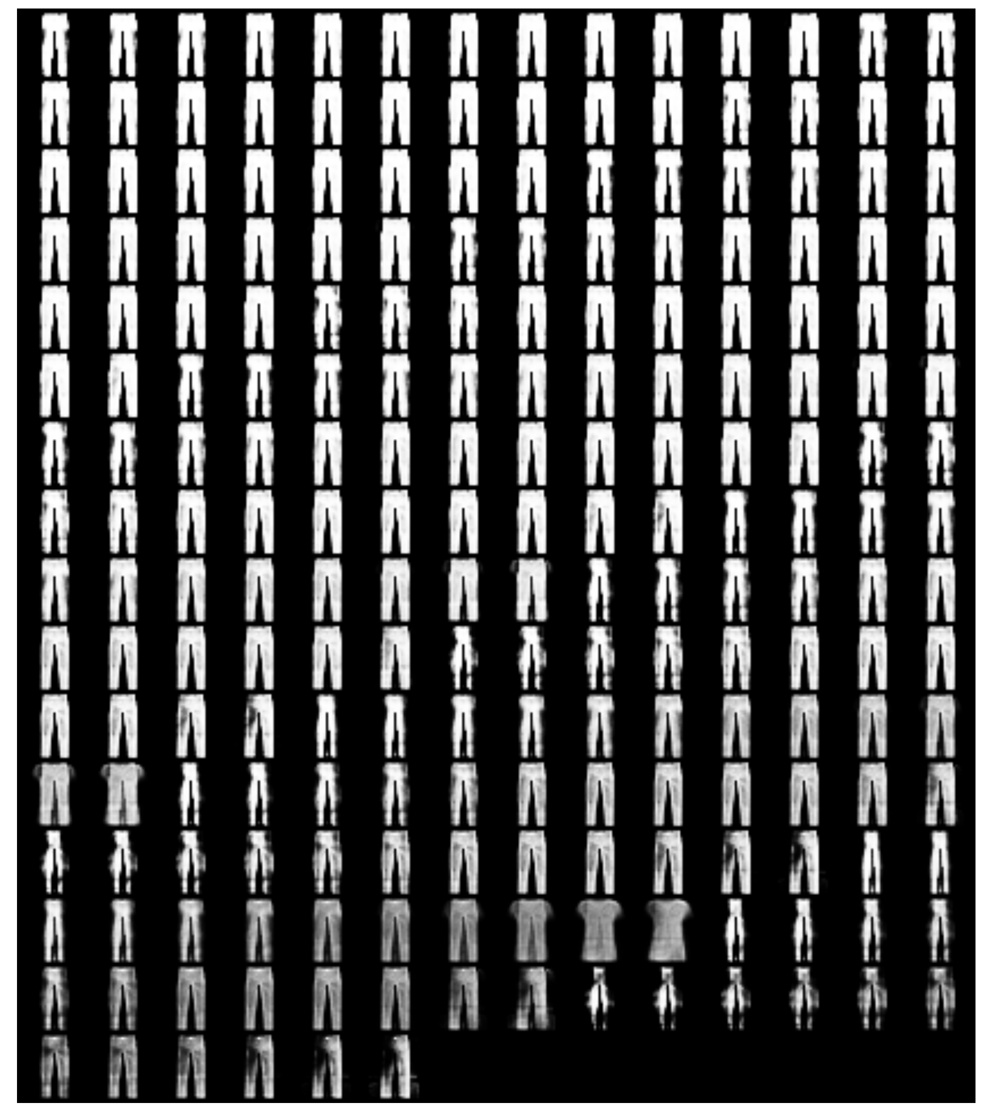
A SIMPLE VAE EXPERIMENT

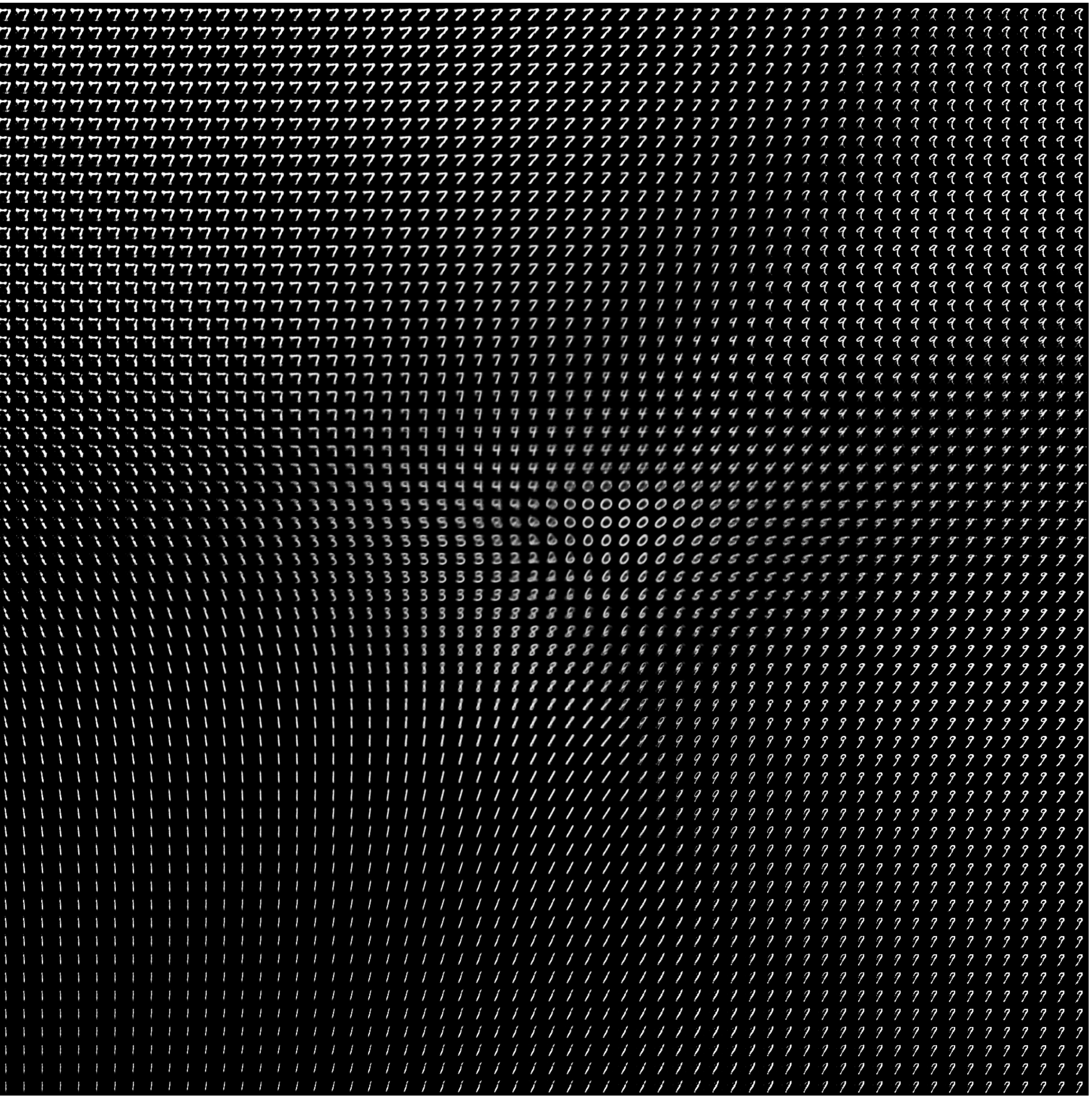
- Very simple setting just to see creativity without meta-learning.
- With a simple VAE
- Manipulate Latent Encoding vectors (size 3)
- 4 shots from digit ‘9’, same 4 samples added to dataset as many times so that almost same number of samples exists.

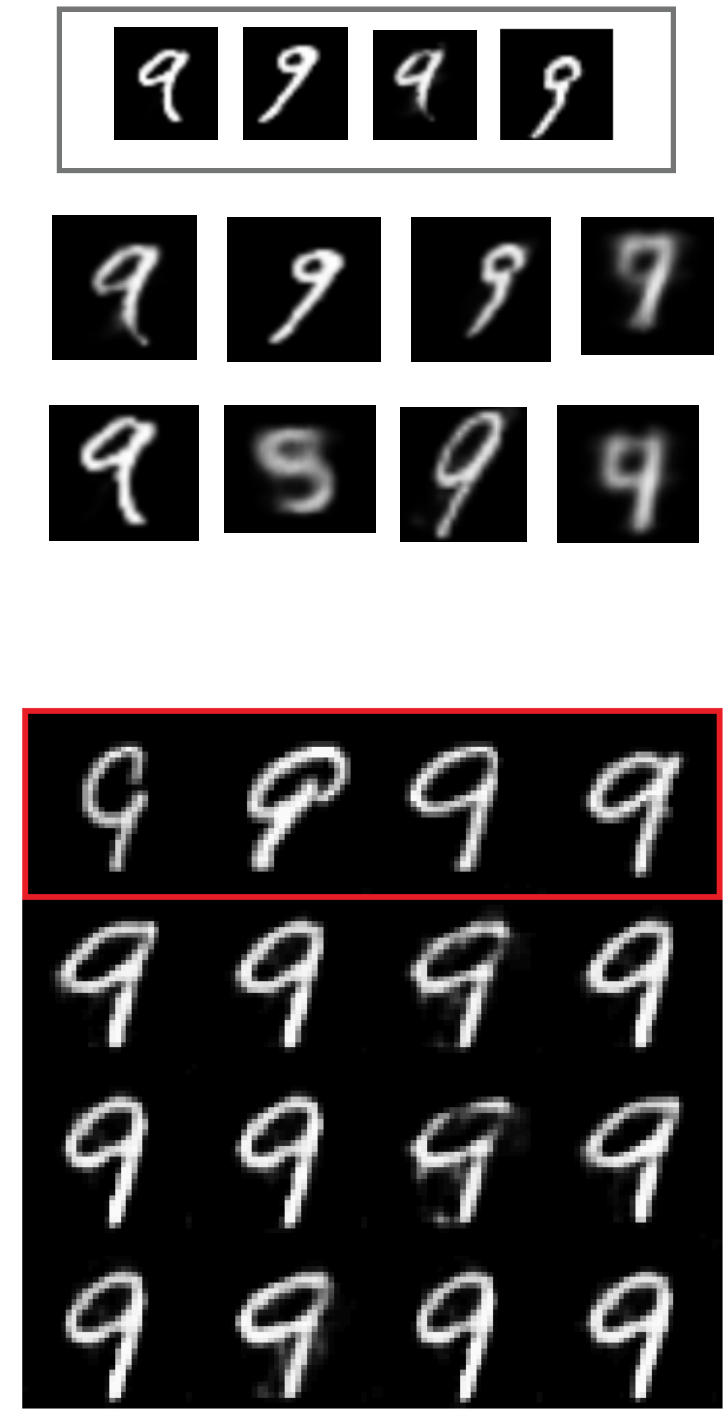
ONE-SHOT PANTS

Since Latent encoded vector is able to manipulate the size and the width of the generated samples, even we added the same sample many times we are able to generate pants from many sizes and widths, even width different shades.









MNIST; 50,000 update; 10 gradient steps

CONCLUSION

- We explored the latent space on both synthetic data(8 Gaussians) and images.
- Meta-Learning has promising results on generalizing on new samples and learning key features of the images very quickly during fine-tuning.
- However, still computationally expensive and generated samples lack novelty and creativity.
- How to evaluate creativity is still unclear.

FUTURE WORK

- Experimenting using meta-learning with simpler settings on wide variety of GAN architectures and VAEs
- Generating multi channel images and larger images, which is also not explored in the recent papers.

Thanks:)

Q & A

Güneş Yurdakul