

MVM+ : 메트릭 러닝을 이용한 생성모델 연구

[메트릭 공간에서의 적대적 생성]

MVM+ : Generative Model via Adversarial Metric Learning

* 1 이재원

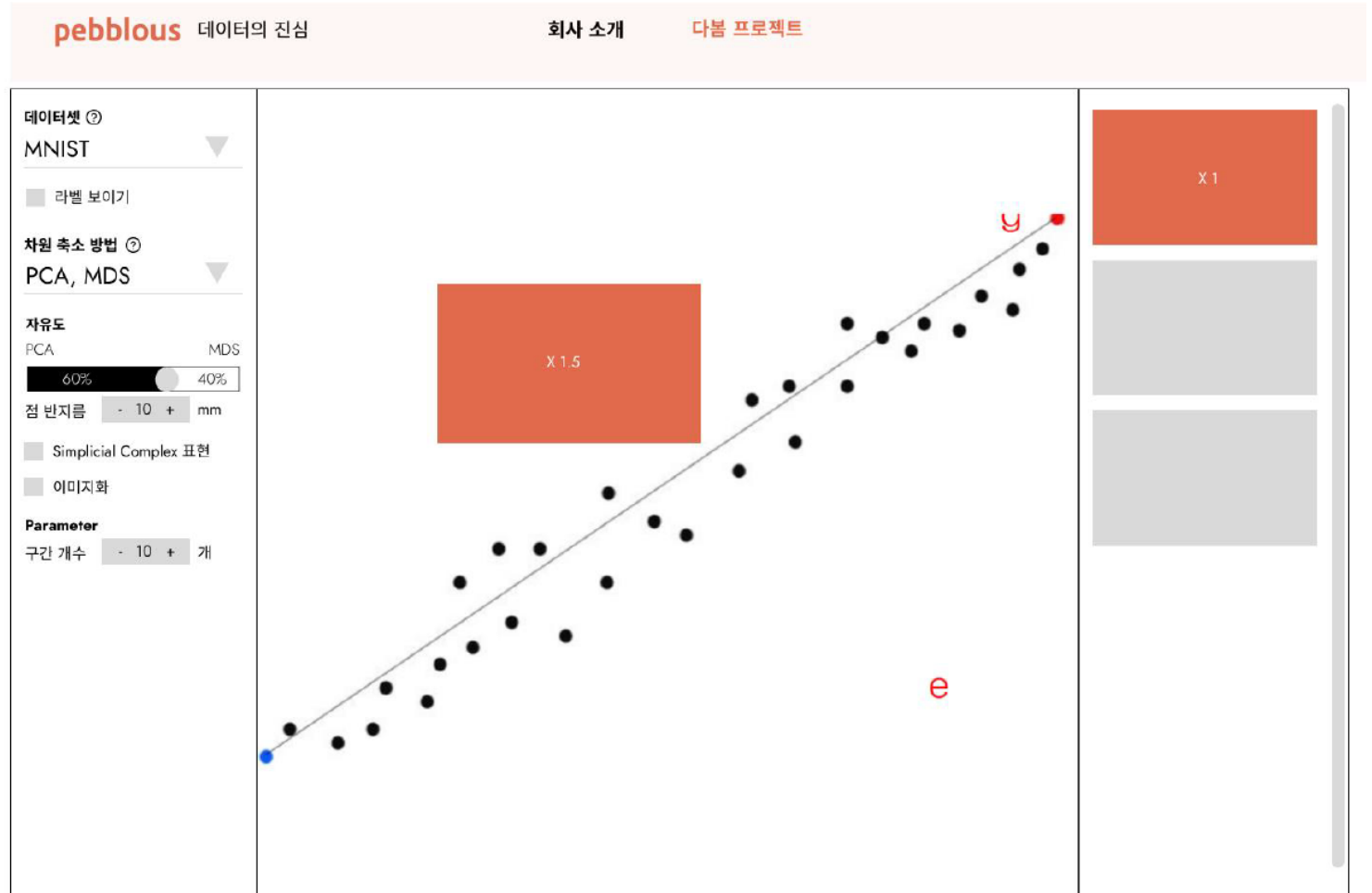
¹ KAIST 전산학부

*발표자(jwlee3746@kaist.ac.kr)

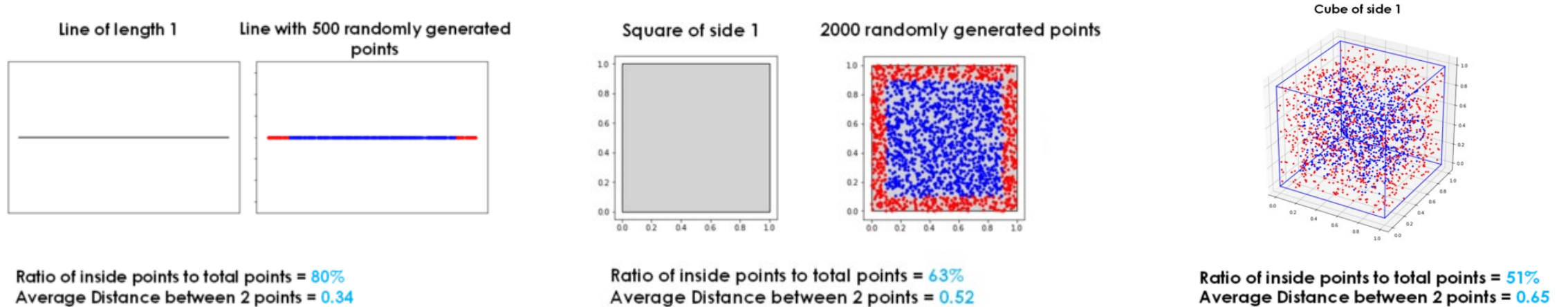
pbls

Problem Setting

- 데이터의 모양과 특성을 보여주는
〈데이터 시각화〉
- 데이터 품질을 평가하고 개선하는
〈데이터 클리닉〉
- 데이터 부족에 정밀하게 생성하는
〈데이터 타겟팅〉



Curse of Dimensionality

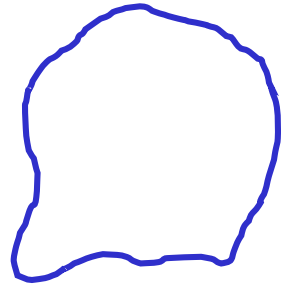


1. 차원이 커질수록 데이터가 희소(sparse)해진다
 - 더 많은 랜덤 샘플들이 공간의 가장자리에 만들어진다
2. 거리 metrics (Euclidean, Manhattan, etc)가 의미를 잃어버린다
 - 두 샘플 사이의 거리 값이 커지면서 서로 유사해진다

Deep Metric Learning

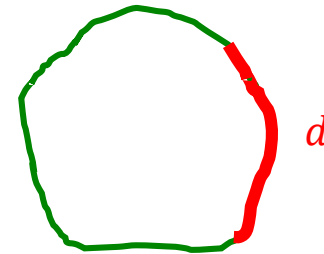
R^D (고차원 데이터 공간)

M



R^n (저차원 메트릭 공간)

N



1. 데이터가 분포하는 R^D 에서는 샘플 사이의 거리가 의미를 갖지 않는다 (차원의 저주)
2. 미분 다양체 M, N 사이의 임베딩 $f : M \rightarrow N$ 생각
 - Euclidean loss (contrastive, triplet)를 최적화하는 과정에서 metric h 를 리만 metric으로 갖는 manifold N 생성
 - N의 리만 metric h 의 pullback을 통해 M의 리만 metric g 를 학습: $g_p(v, w) := h_{f(p)}(df_p(v), df_p(w))$
3. 따라서 Metric Learning이란, 데이터 공간 R^D 에서는 특정할 수 없었던 샘플 간의 관계 혹은 Metric 을, 차원 축소된 공간 R^n 에서 리만 manifold (N, h) 를 학습함으로써 특정하는 과정이라고 볼 수 있다.

MVM

GAN vs MVM *

Differences	GANs	MvM
Main point of view	statistics	geometry
Matching terms	means, moments, etc.	centroids, p -diameters
Matching criteria	statistical discrepancy	learned distances
Underlying metric	default Euclidean	learned intrinsic
Objective functions	one min-max value function	two distinct objectives

GAN

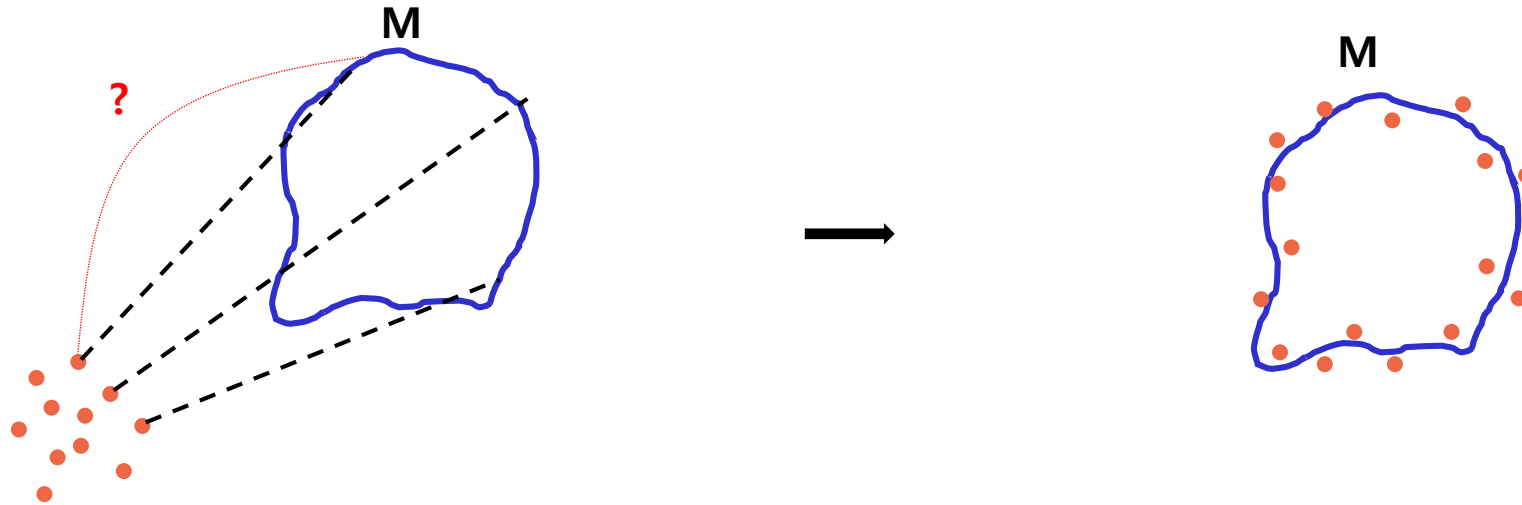
- 통계학적 접근
- 데이터 시각화 어려움

MVM

- ICCV 2021 발표 (Microsoft)
- 기하학적 접근
- 생성과 동시에 데이터 시각화 가능

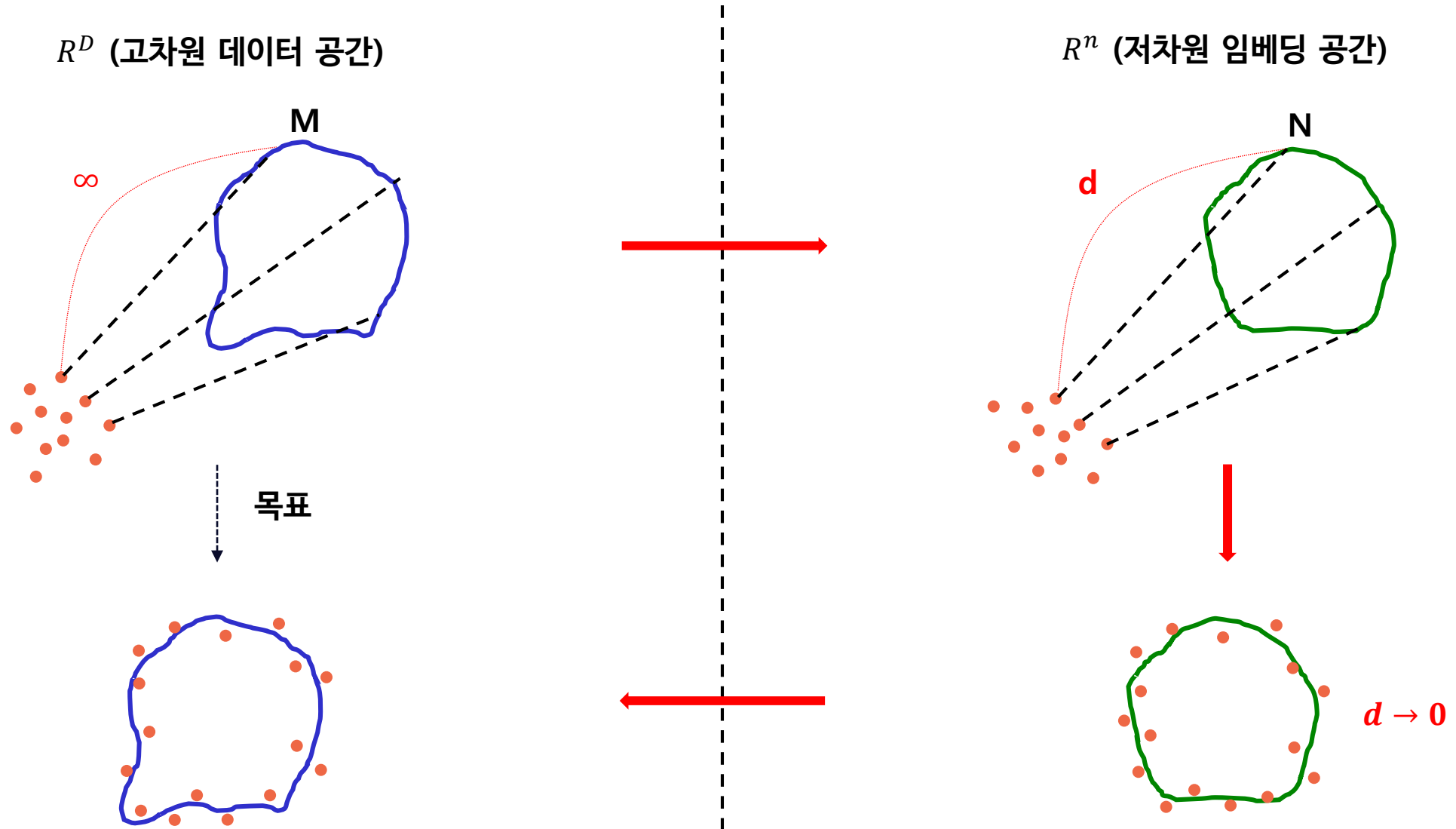
Goal of MVM

R^D (고차원 데이터 공간)



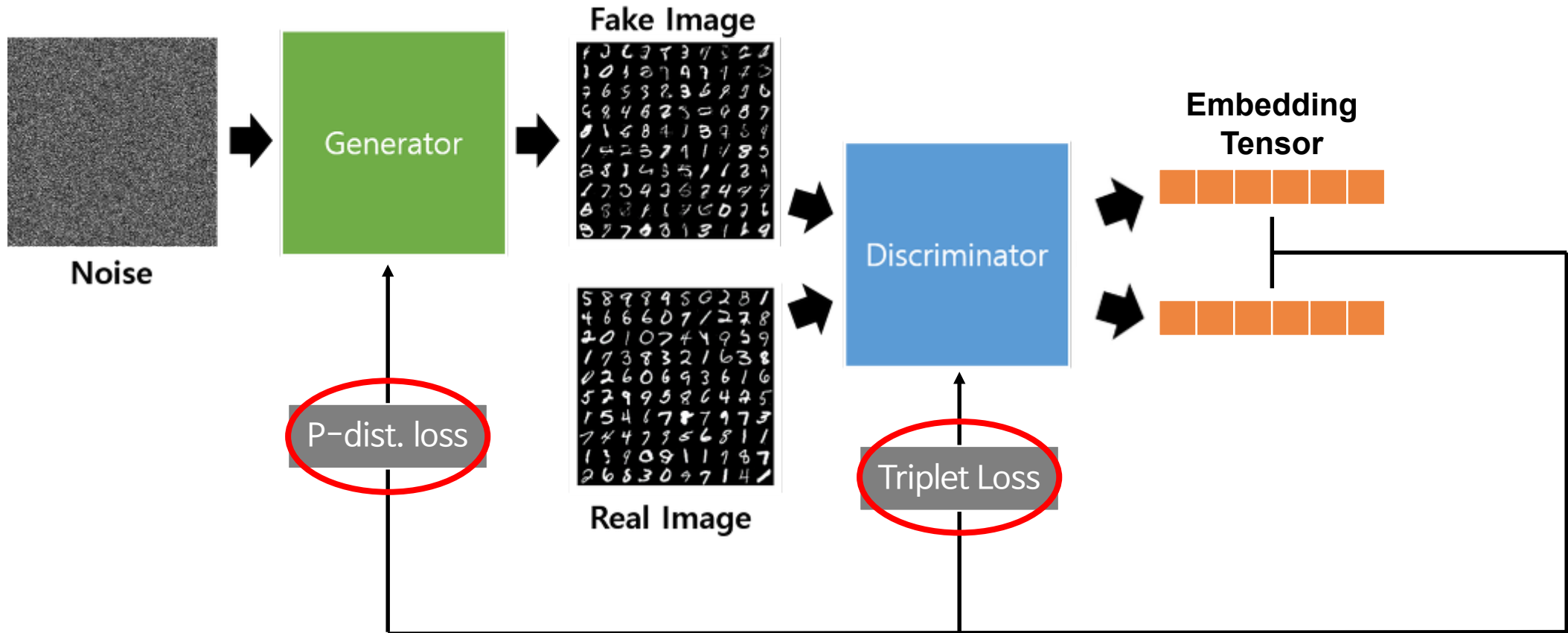
1. 가정 : 데이터가 고차원 공간에서 특정 manifold를 형성
2. 목표 : 실제 데이터 manifold M 에 **condensed** 하게 만들어지는 것을 목표로 함

Deep Metric Learning in MVM

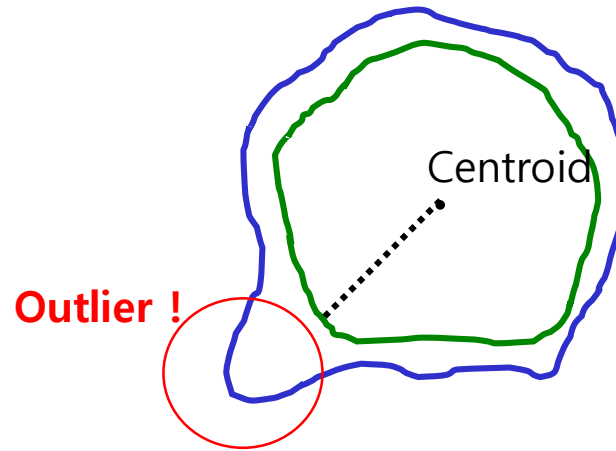


Related Work

MVM pipeline

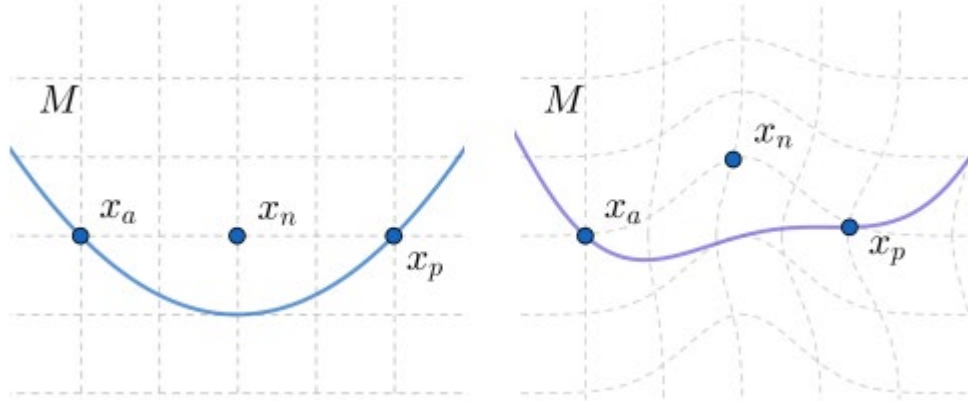


Manifold Matching Loss



- 실제 데이터 manifold M 의 모양을 M 의 sample points 를 통해 추정
 - M 의 중심 → Frechet mean
 - M 의 크기 → p -diameter
- 문제점
 - M 의 모양을 단순히 frechet-mean(centroid) 와 p -diameter로 가정 → MVM의 취약점 1
 - M 의 모난 부분 (Outlier) 을 전혀 반영하지 못한다
 - 만약, M 이 Swiss roll 처럼 독특한 형상을 띄고 있다면?

Triplet Loss



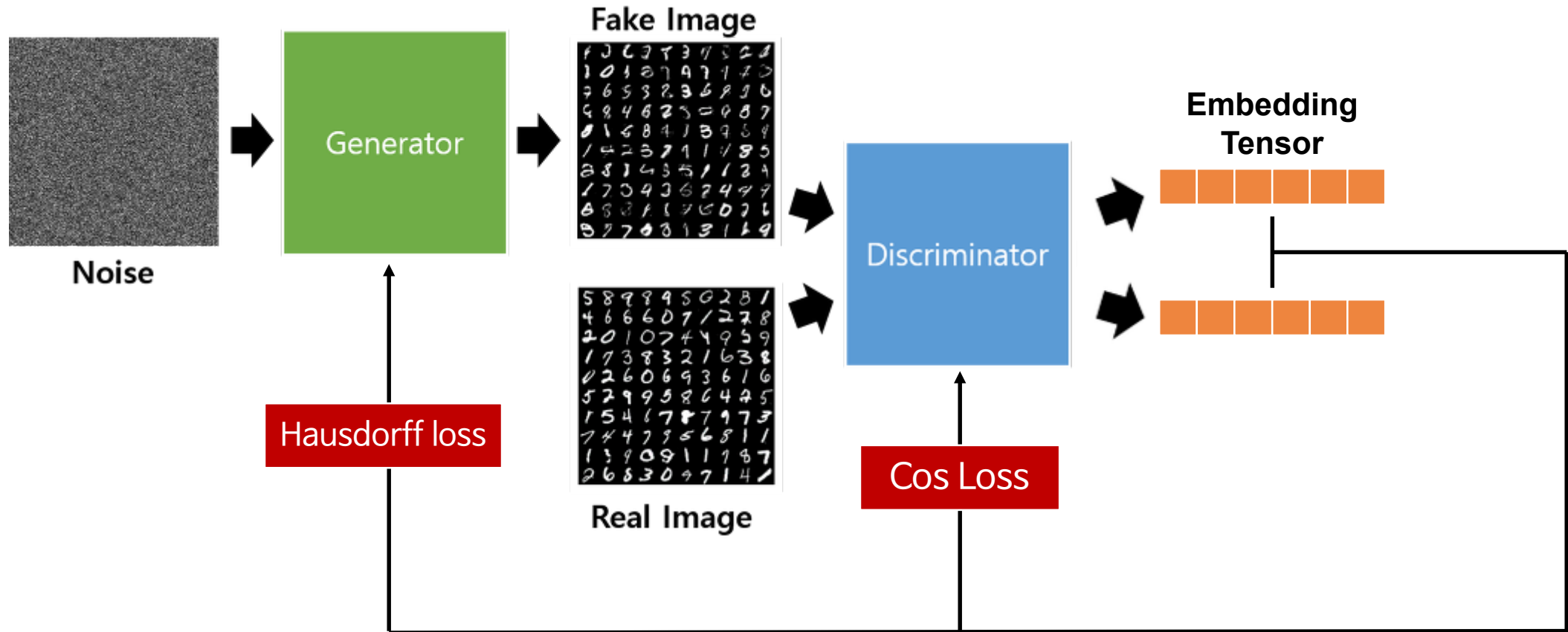
성능 차이 X

$$L_{apn} = \max\{0, \cancel{d^2(x_a, x_p)} - \cancel{d^2(x_a, x_n)} + \alpha - \gamma \cos(g_w(x_n) - g_w(x_a), g_w(x_p) - g_w(x_a))\};$$

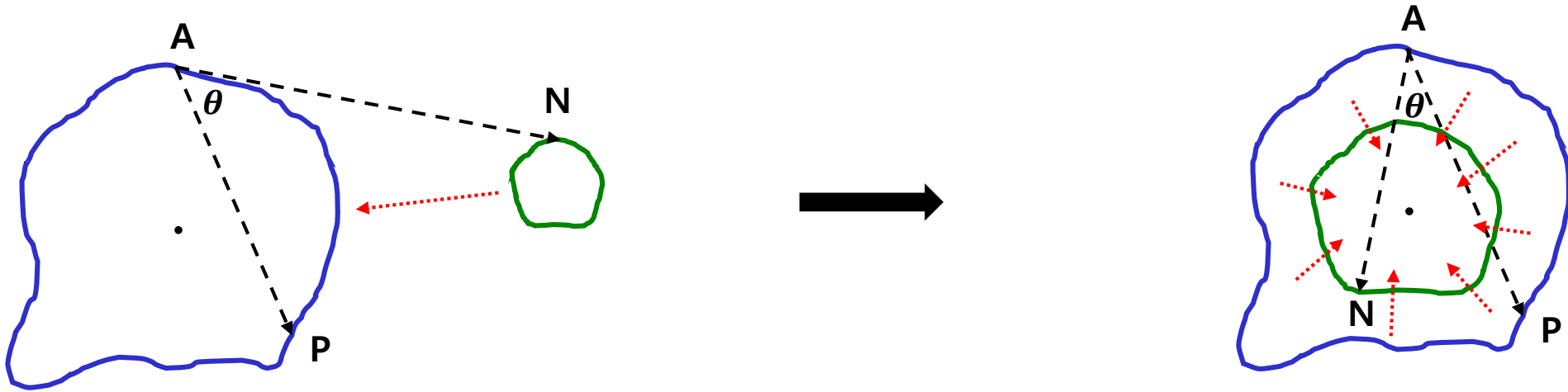
- Triplet loss가 manifold의 등근 부분을 평평하게 해줄 것이라고 가정
- 평평한 manifold를 만들 수 있다면, 측지선 거리 = 유클리디언 거리
- 문제점
 - Triplet loss가 manifold를 평평하게 해준다는 가정은 옳지 못함 → [MVM의 취약점 2](#)
 - 실제로 학습 과정에서 triplet loss의 distance 부분이 아닌, cosine similarity factor가 중요하게 작동함

MVM+

MVM+ pipeline



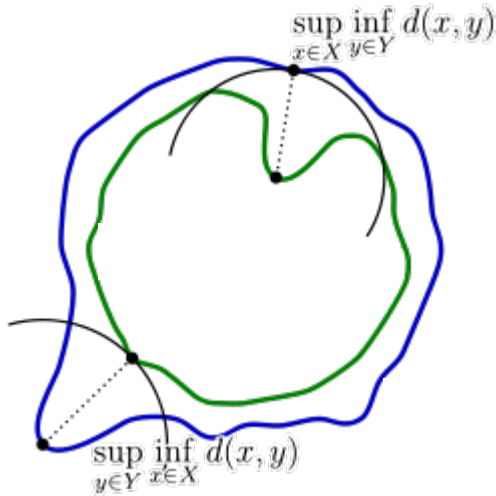
Cosine Similarity Loss



$$L_{apn} = -\text{Cos}(g_w(x_n) - g_w(x_a), g_w(x_p) - g_w(x_a))$$

- Real 데이터 manifold의 두 샘플 anchor, positive 를 지정, 생성 데이터 한 샘플을 negative로 지정
- 지정 세 샘플 사이의 각도를 좁힌다
- 효과
 - 데이터 manifold 형상을 유지
 - Generated 데이터 manifold를 Real 데이터 manifold 중앙에 포함

Average Hausdorff Distance



$$\tilde{\delta}_H(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

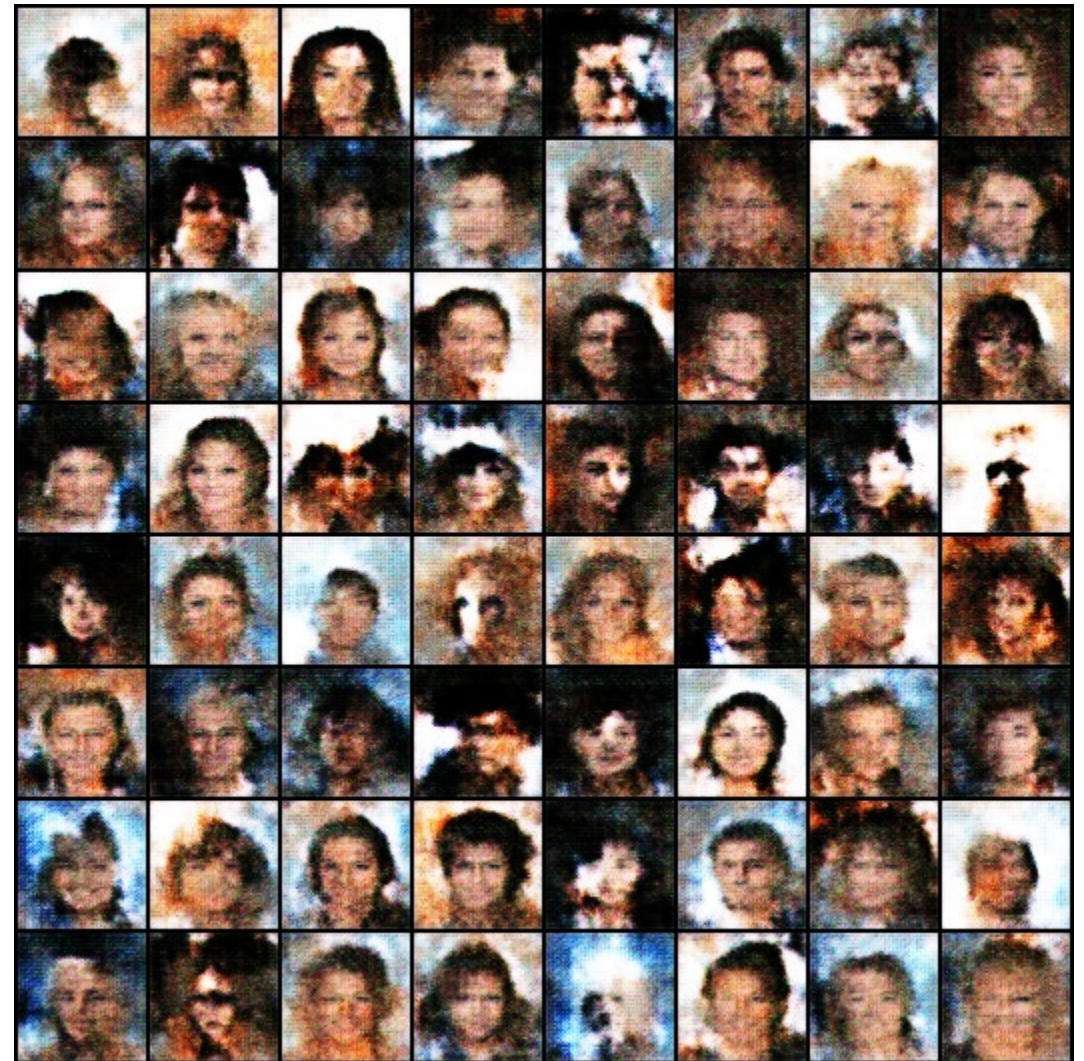
$$d_{AH}(X, Y) = \frac{1}{|X|} \sum_{x \in X} \min_{y \in Y} d(x, y) + \frac{1}{|Y|} \sum_{y \in Y} \min_{x \in X} d(x, y)$$

- Manifold matching을 위해 hausdorff distance 사용
- 실제 데이터의 manifold X에 condensed 한 생성 데이터 매니폴드 Y를 만듦
- 일반적인 Hausdorff distance는 Outlier에 취약하기 때문에 Average Hausdorff distance로 변환

Early Experiments

MVM

- MVM
- 이미지 생성
- 초기 설정 그대로 실험 결과 : 결과 bad



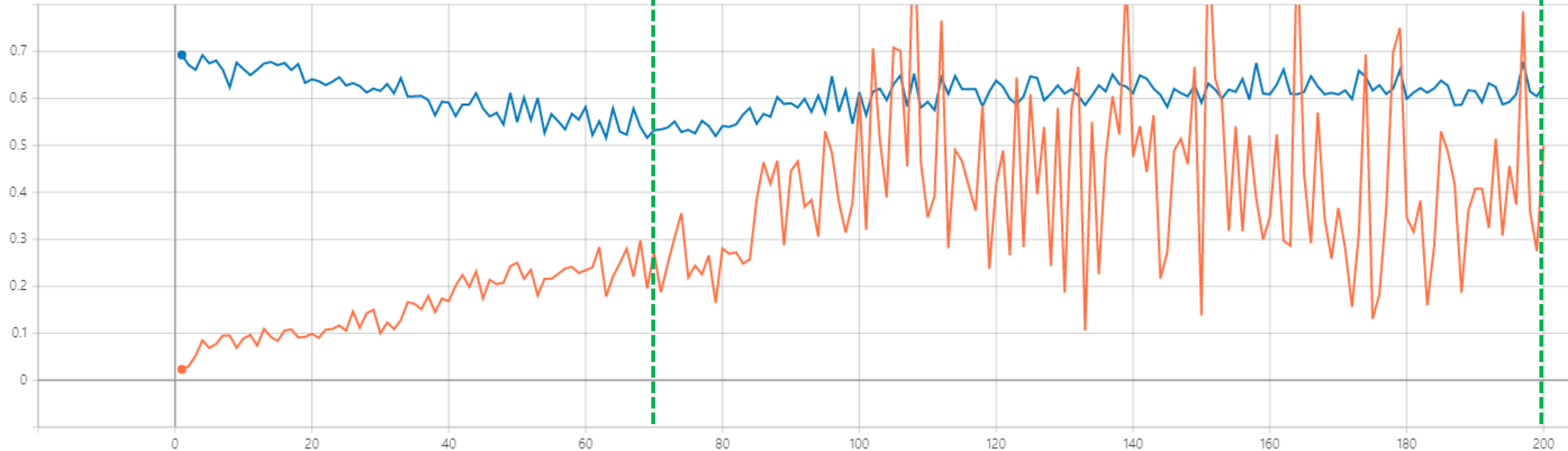
Experiments

MVM

- MVM
- Loss가 불안정함
- Collapse 문제 지속 발생



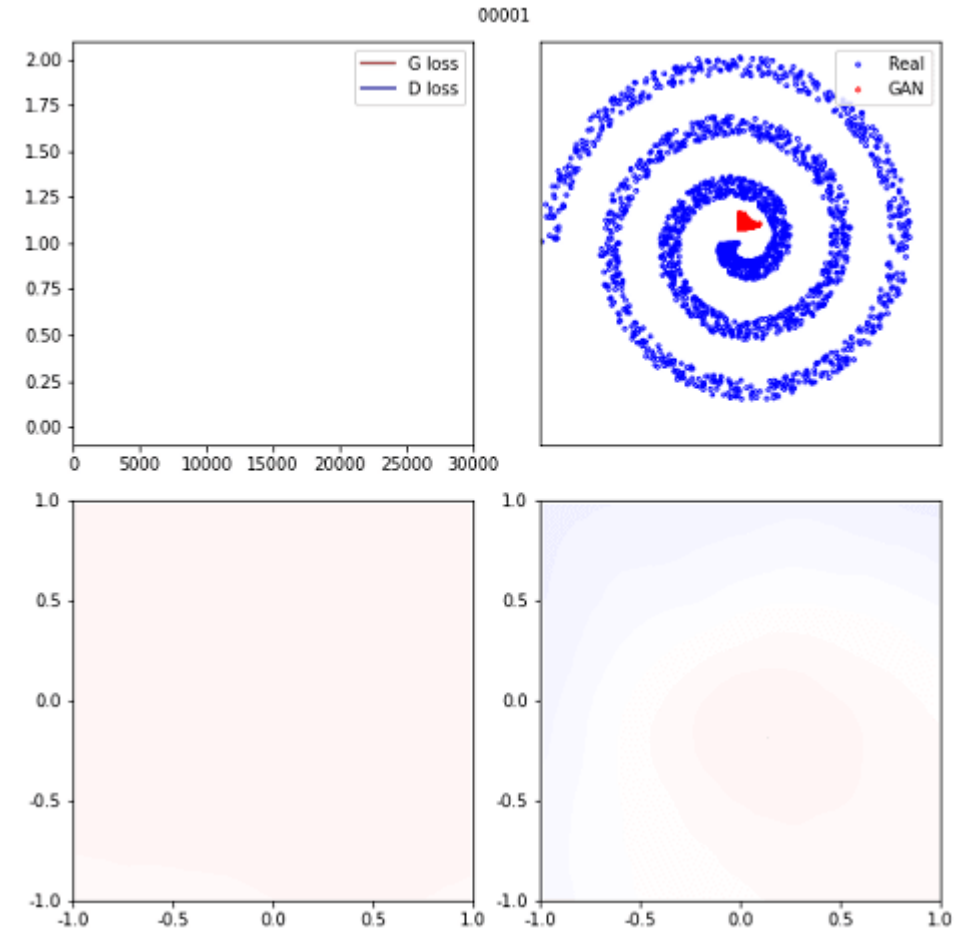
train
tag: loss/train



Experiments

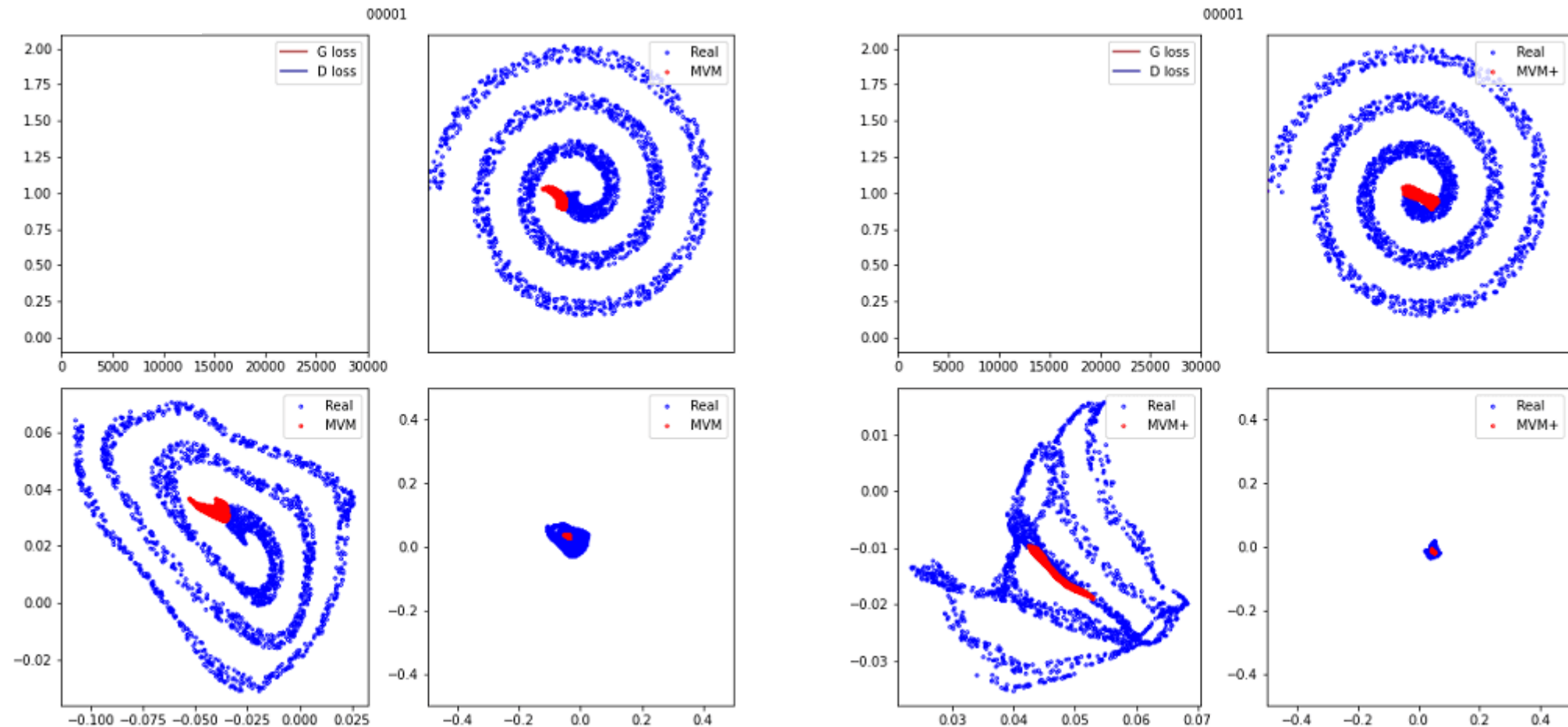
Synthetic Data Generation

- GAN
 - 2차원 가상 데이터 (spiral)
1. Top Left : loss
 2. Top Right : Real and Fake Samples (manifold)
 3. Bottom Left : Latent space confidence
 4. Bottom Right : Data space confidence



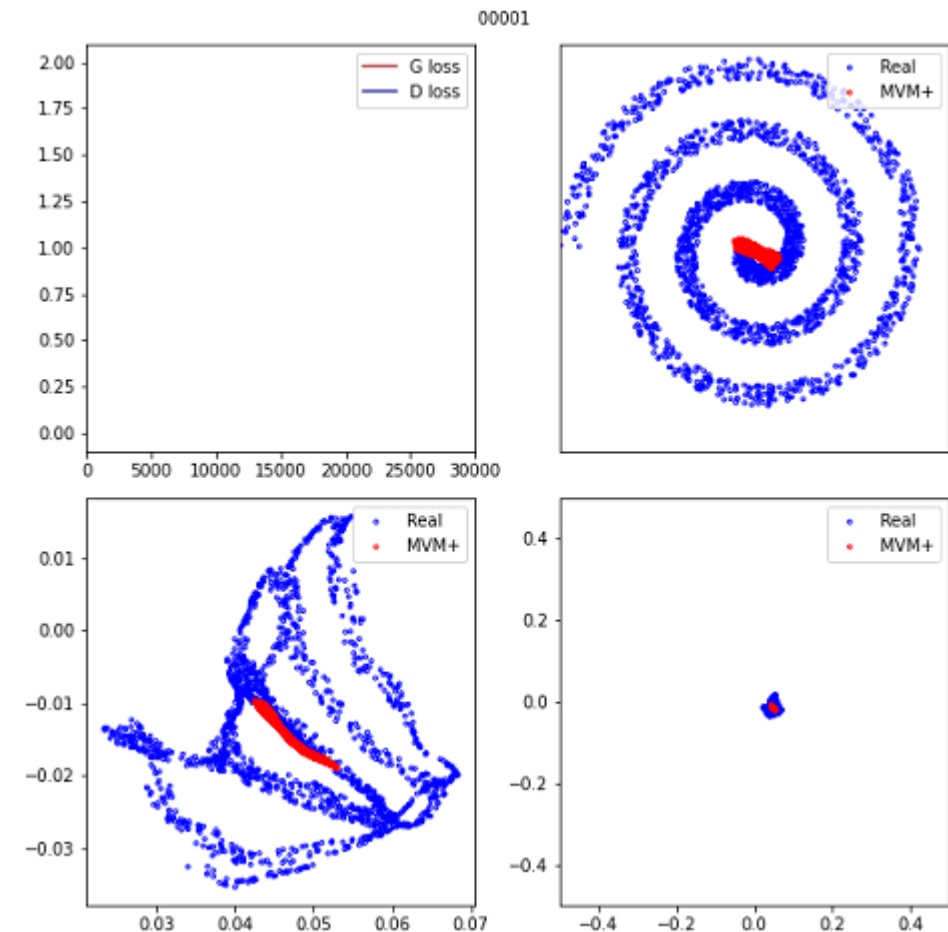
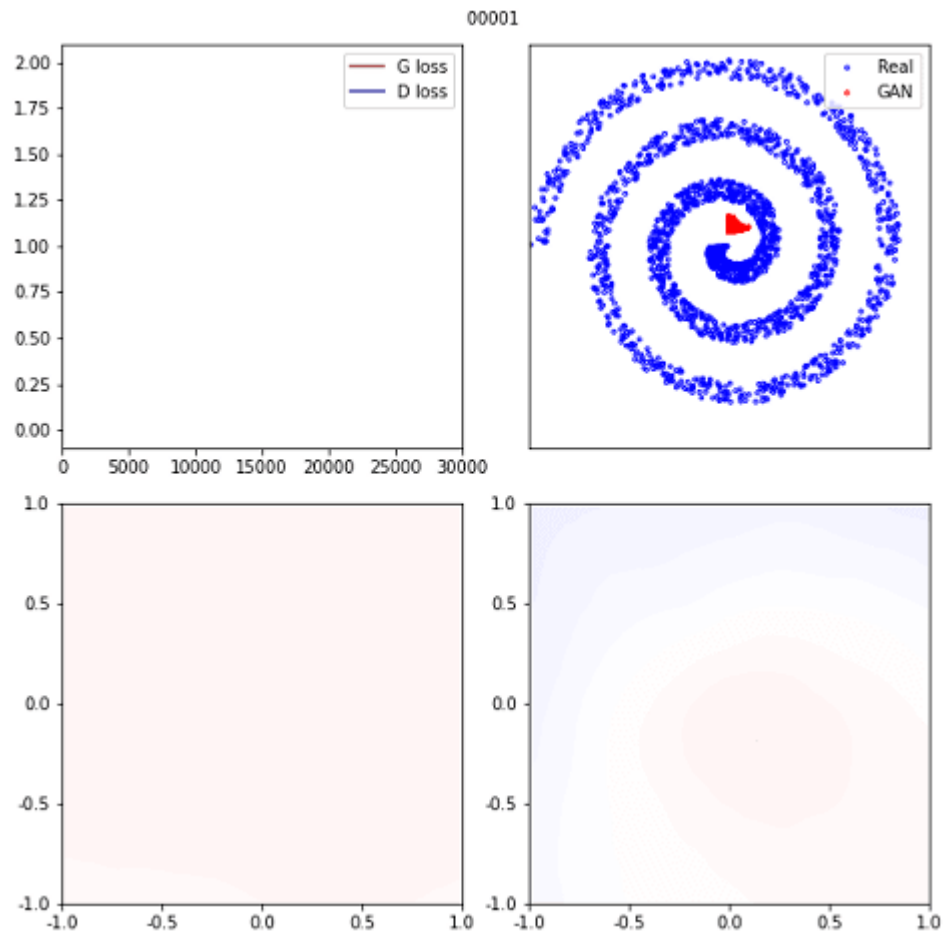
Synthetic Data Generation

- MVM vs MVM+



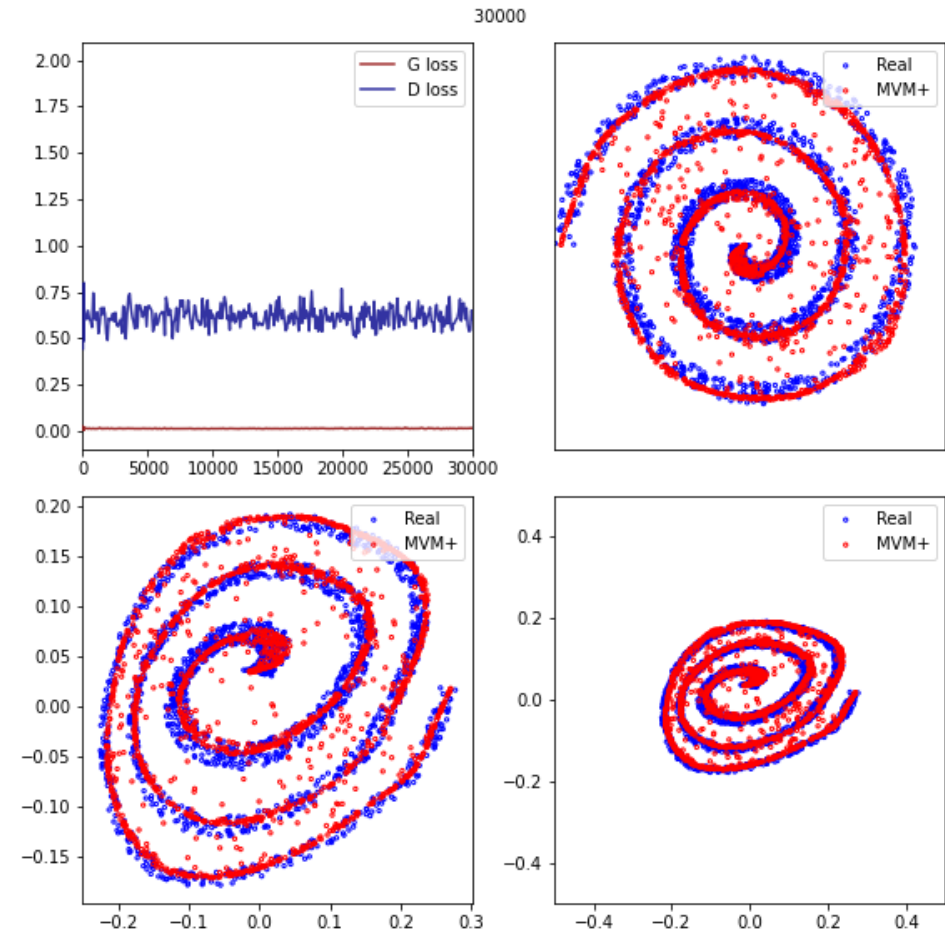
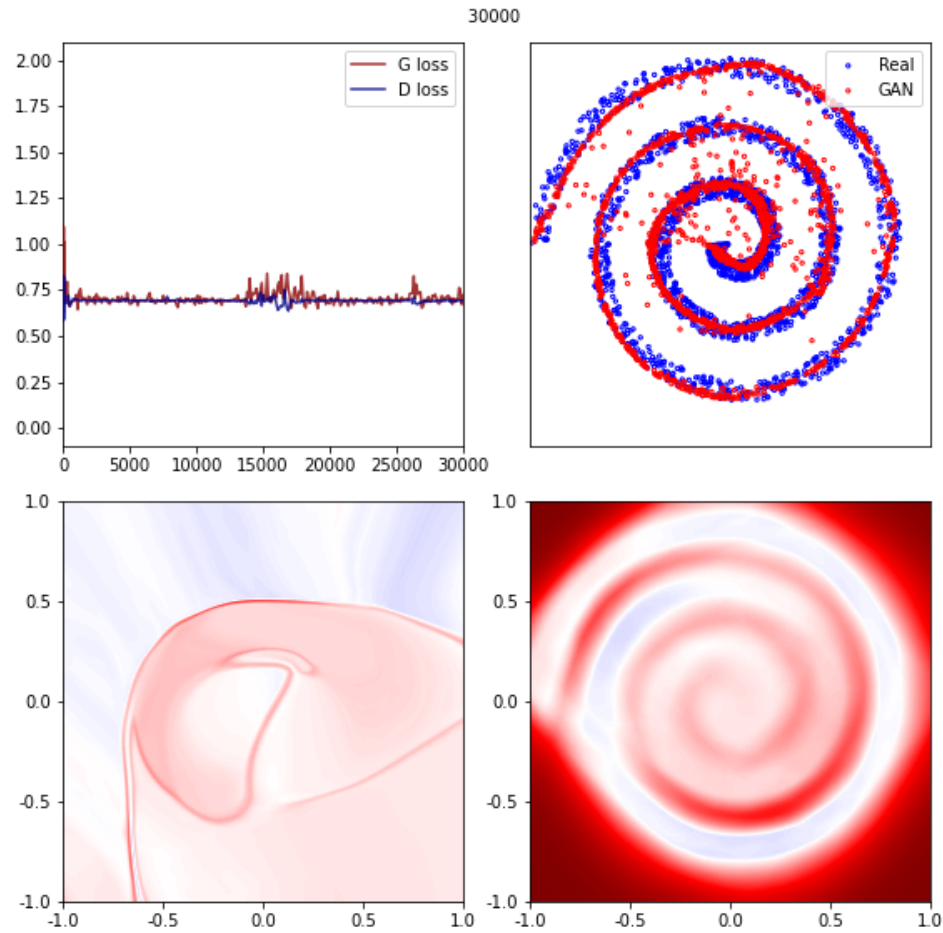
Synthetic Data Generation

- GAN vs MVM+



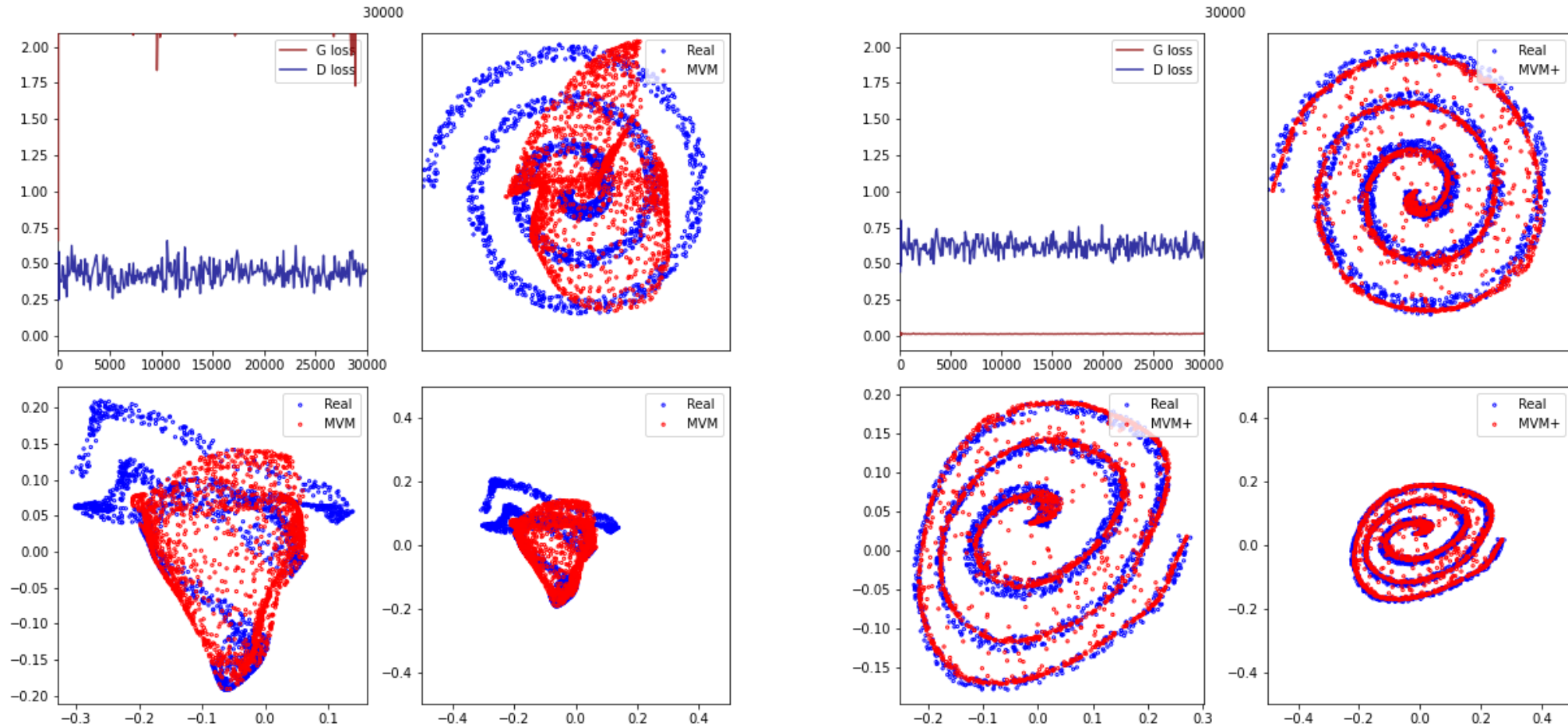
Synthetic Data Generation

- GAN vs MVM+



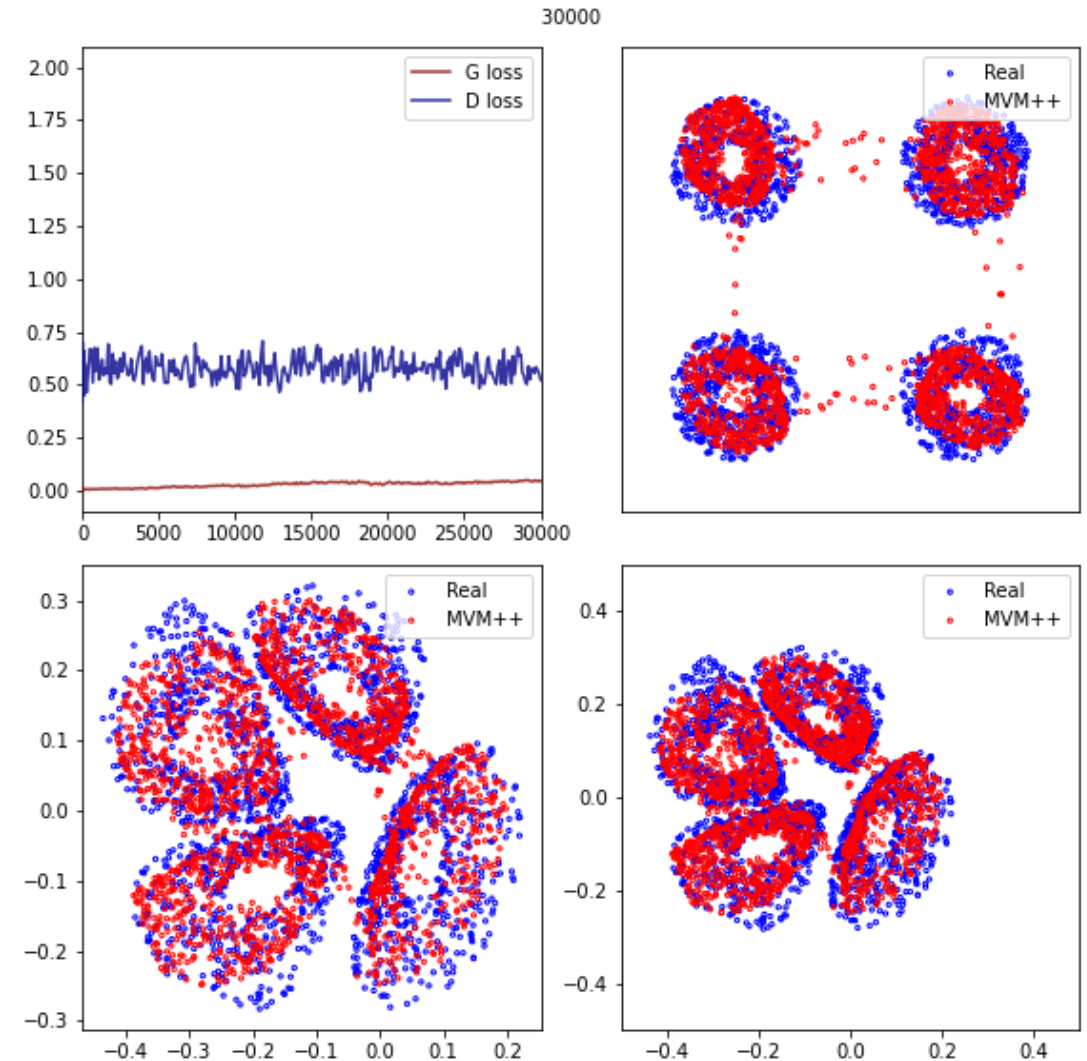
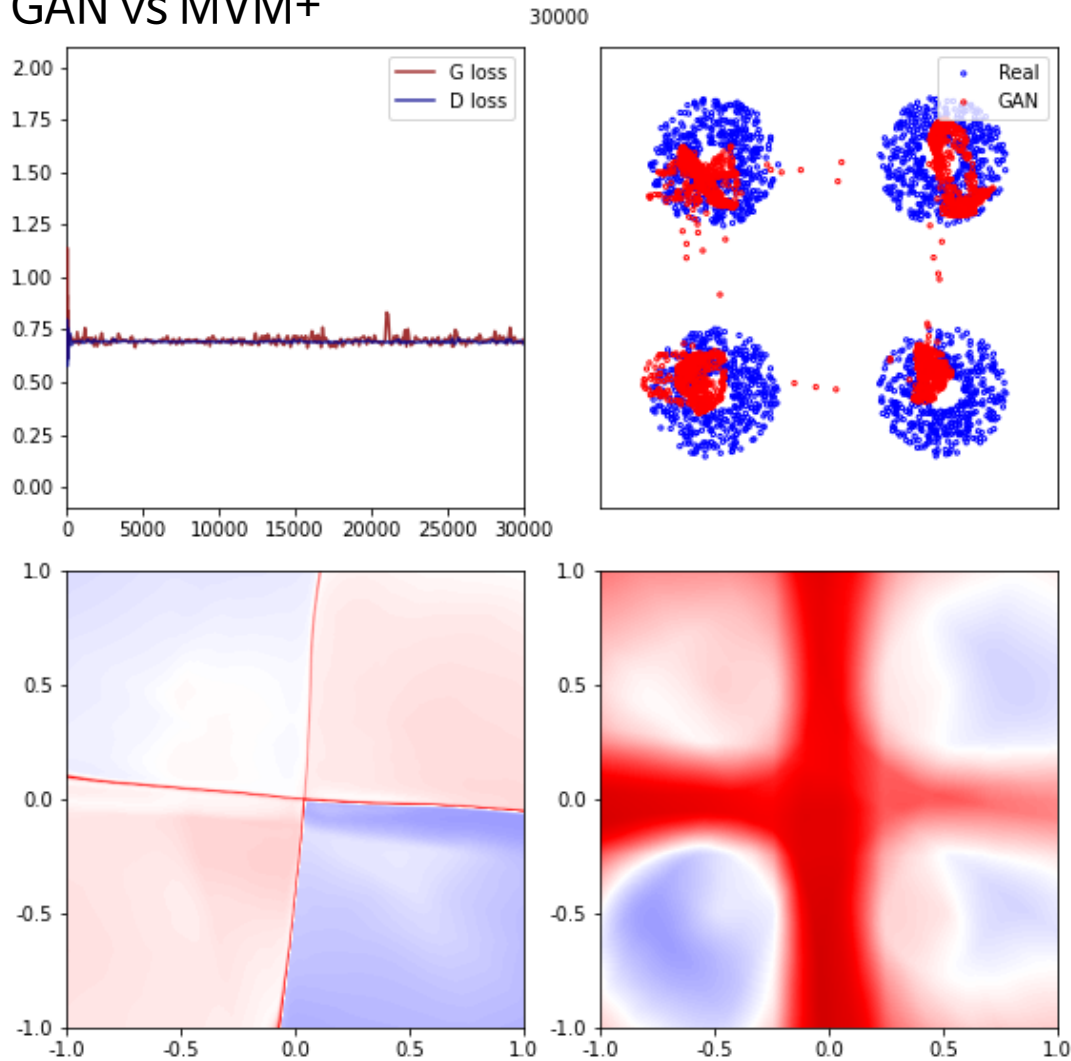
Synthetic Data Generation

- MVM vs MVM+



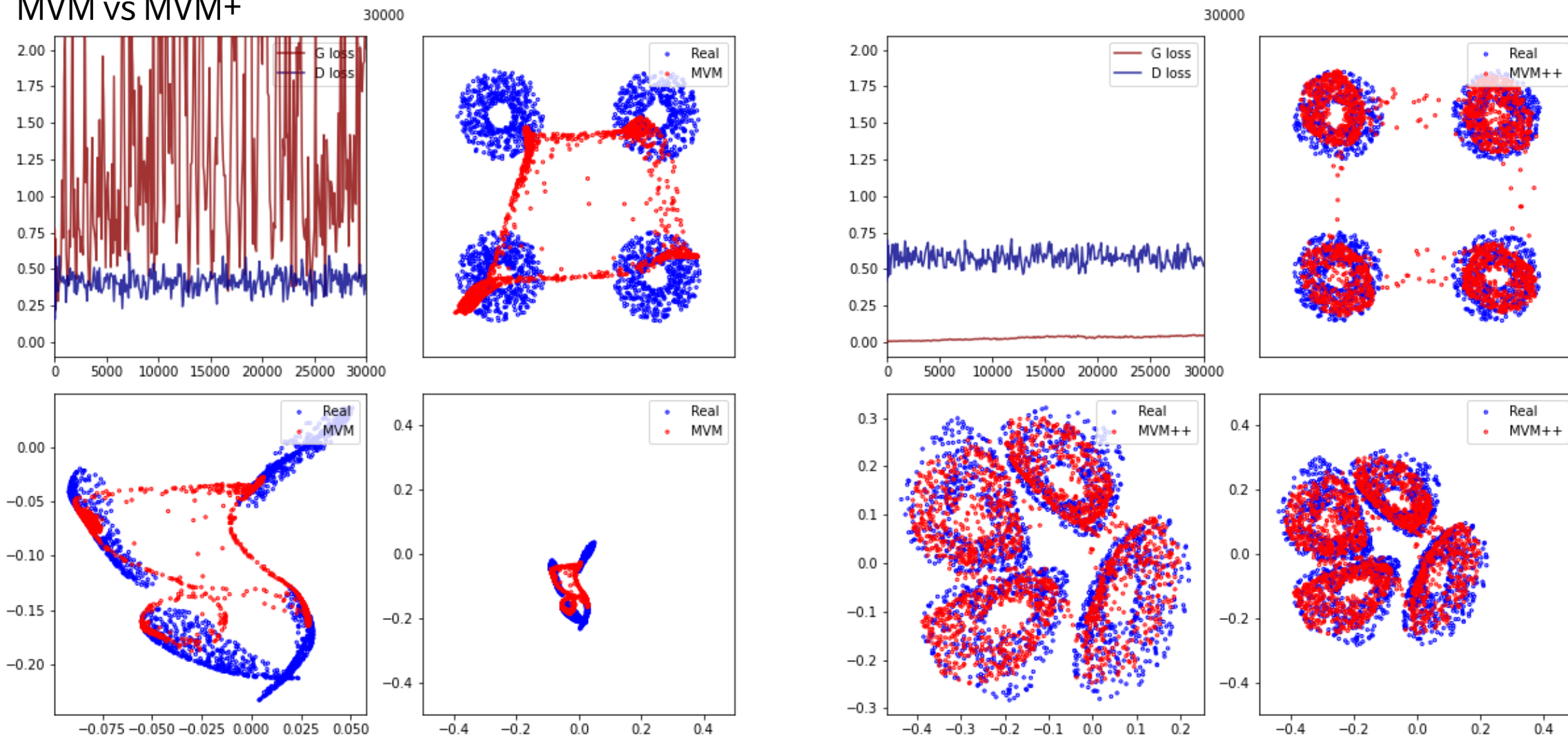
Synthetic Data Generation

- GAN vs MVM+

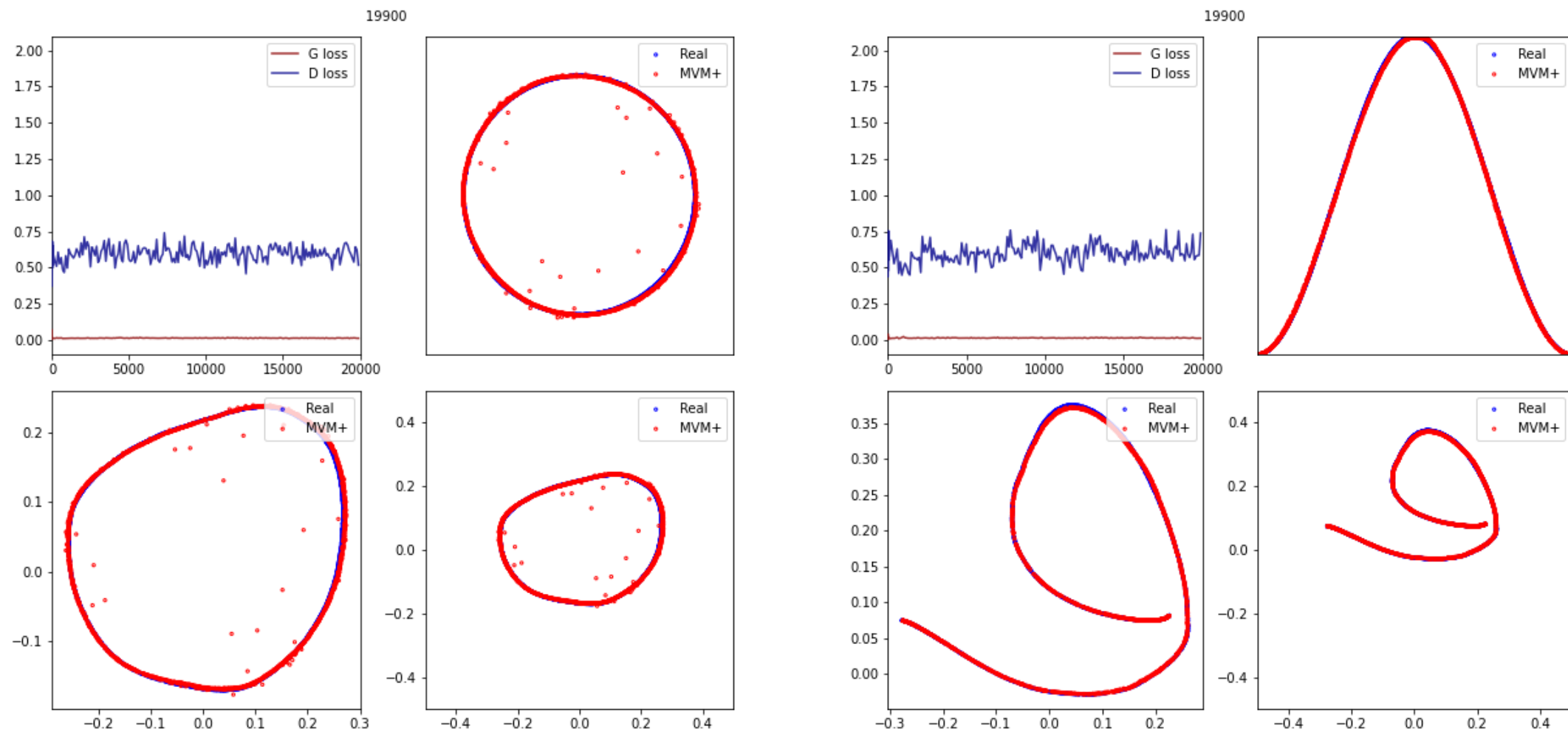


Synthetic Data Generation

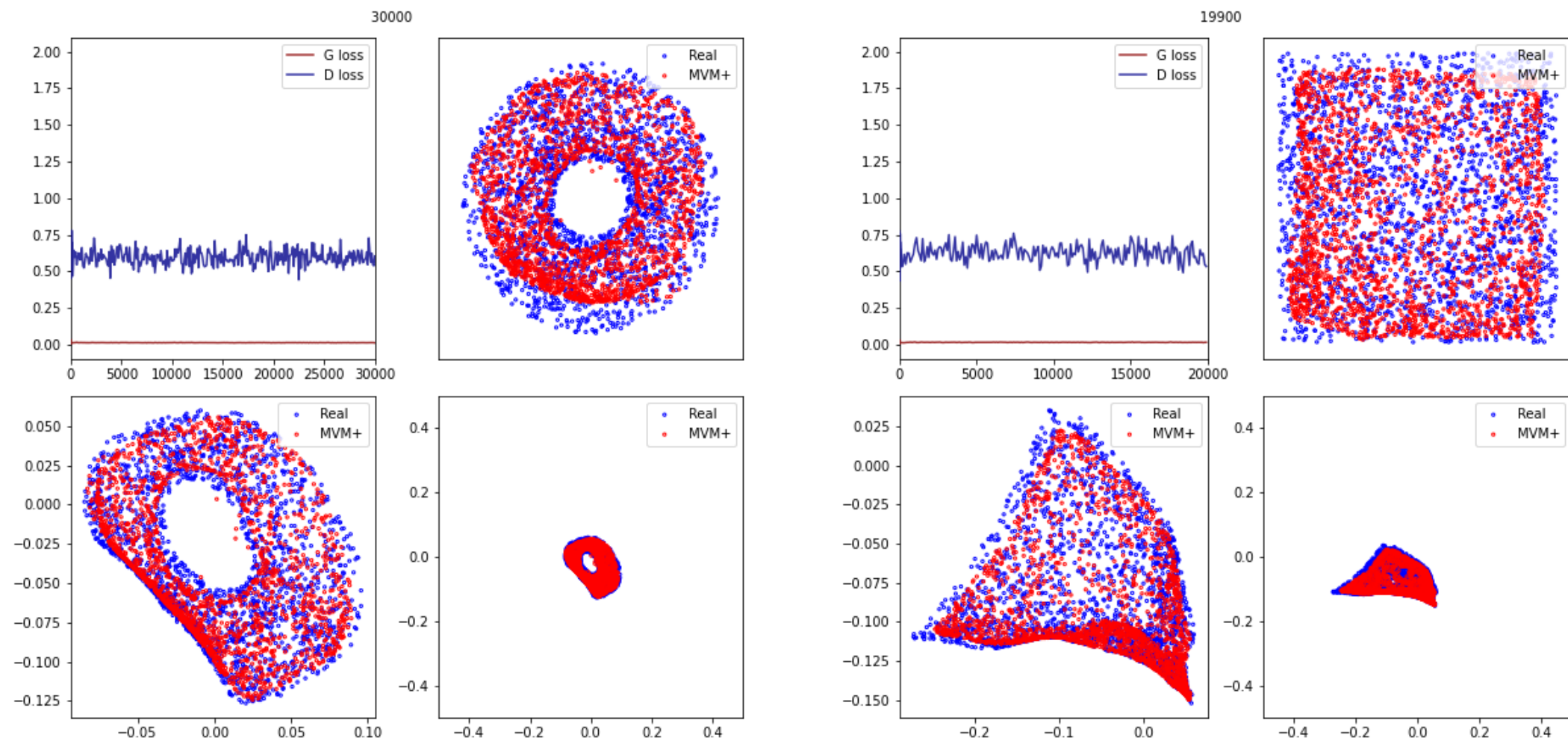
- MVM vs MVM+



Synthetic Data Generation



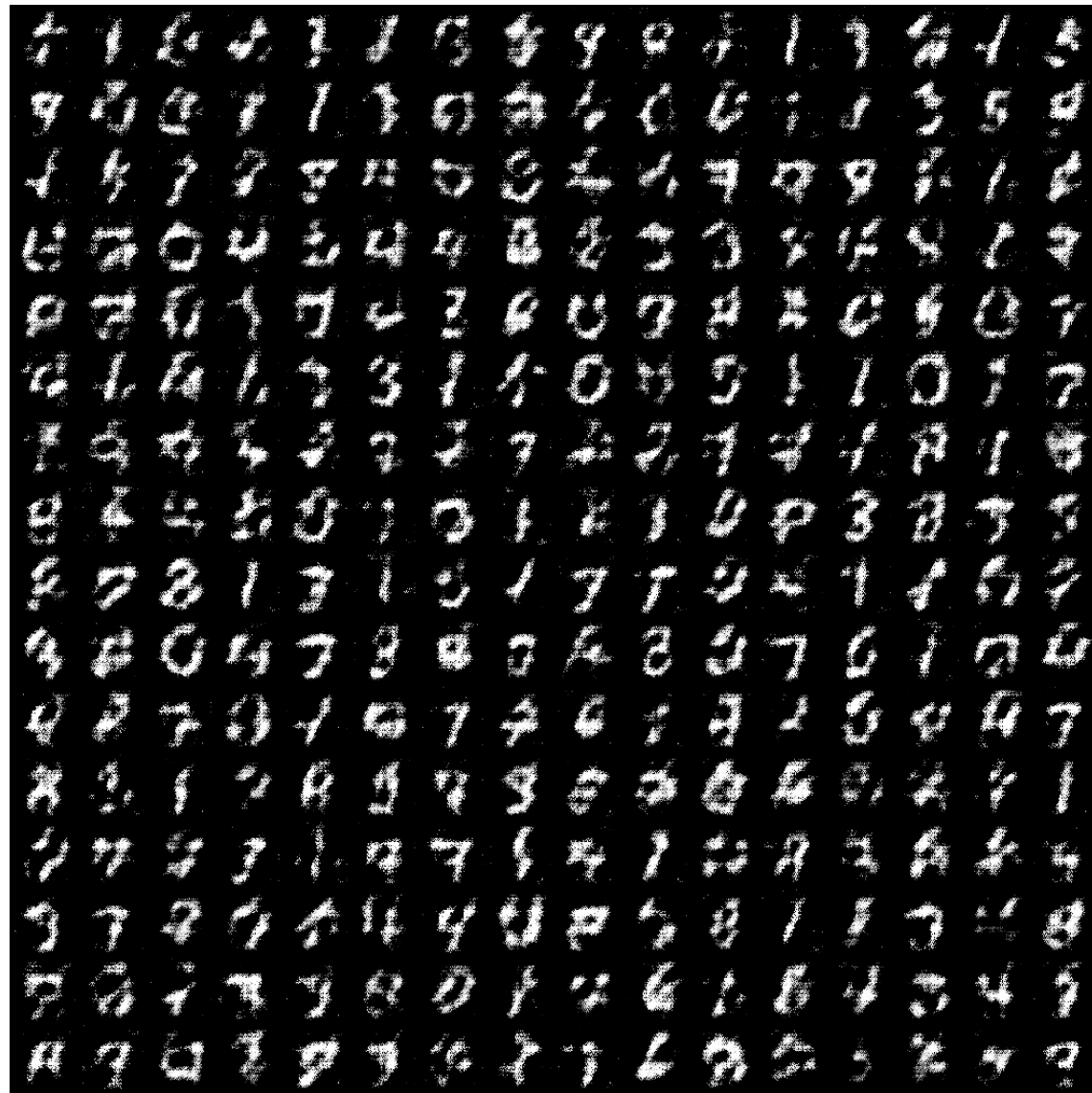
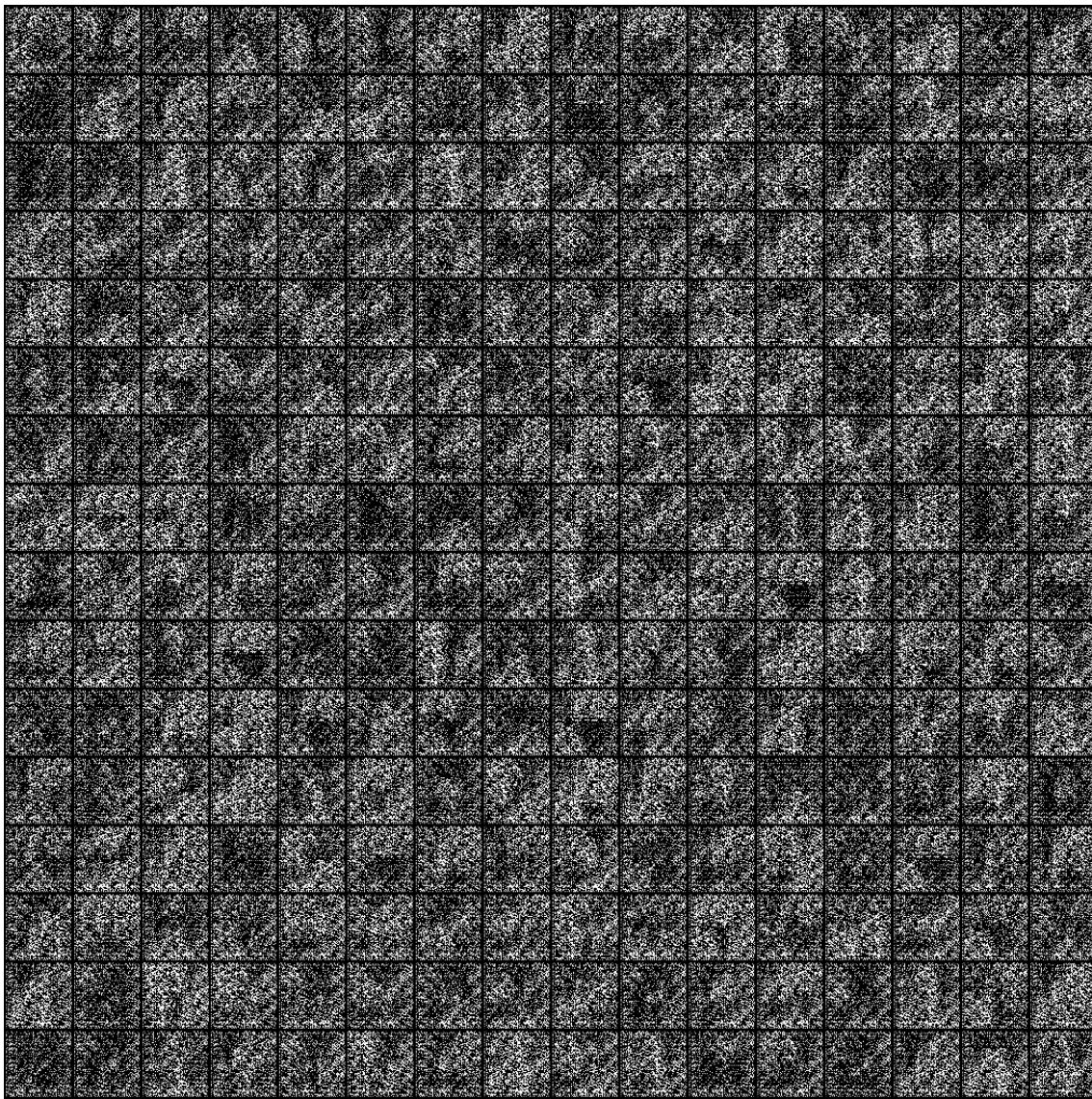
Synthetic Data Generation



Key Results

Results

MVM vs MVM+ (epoch 1)



Results

MVM vs MVM+ (epoch 100)

5 2 6 9 7 9 5 9 1 6 1 2 7 4 5 8
2 7 1 9 7 7 9 8 8 6 8 1 5 4 6 6
8 0 1 0 2 9 8 6 1 8 0 2 7 2 2 9
2 0 3 4 6 2 3 3 1 6 6 1 9 5 8 9
9 1 8 6 2 7 1 0 6 2 9 1 6 2 0 1
9 6 1 3 2 3 7 1 1 7 7 1 9 1 6 5
7 4 6 4 5 9 2 9 8 1 6 2 8 9 5 0
4 8 1 2 7 7 9 9 2 5 8 2 5 5 8 2
6 9 9 9 9 6 9 1 6 1 2 2 7 6 6 9
4 6 7 8 6 6 1 6 8 9 2 3 7 7 2 9
5 6 8 3 1 2 5 3 7 7 8 5 2 5 6 3
7 2 1 6 5 1 7 8 1 9 9 3 2 7 1 2
1 0 5 6 8 7 9 1 7 8 1 5 8 2 1 7
0 7 6 7 6 8 9 9 0 8 2 3 2 6 7 2
2 8 8 1 0 0 4 9 1 4 4 7 0 7 9 3
1 2 6 9 5 6 0 1 1 6 6 9 8 4 5 9

2 0 6 5 6 1 8 1 7 6 9 9 7 5 1 2
5 6 5 4 4 8 9 7 7 6 0 5 0 5 3 8
0 5 4 1 1 0 8 9 9 3 6 7 1 9 9 9
1 2 1 4 8 7 2 9 0 3 5 3 7 2 1
2 2 3 0 0 5 1 1 9 2 7 3 4 8 9 3
9 1 9 4 9 0 8 7 6 8 5 5 0 1 2 5
7 1 7 3 9 5 5 6 3 1 1 6 5 0 7 9
3 2 4 0 2 8 6 4 2 5 2 9 7 2 7
7 6 2 4 2 2 9 2 6 5 0 1 7 8 5 4
0 5 8 1 4 0 2 5 6 2 3 3 6 3 7 5
4 1 5 6 1 4 8 6 2 9 2 1 0 8 4 1
2 8 7 7 4 0 6 6 4 7 6 1 9 9 2 6
4 7 7 4 5 6 3 6 1 5 4 4 3 2 2 9
2 9 2 7 4 4 6 4 3 1 5 0 3 1 8 6
2 1 4 1 7 9 2 7 4 6 7 6 3 0 7 0
5 6 7 3 2 1 3 1 1 1 5 8 2 0 6 2

Results

MVM vs MVM+ (epoch 80)



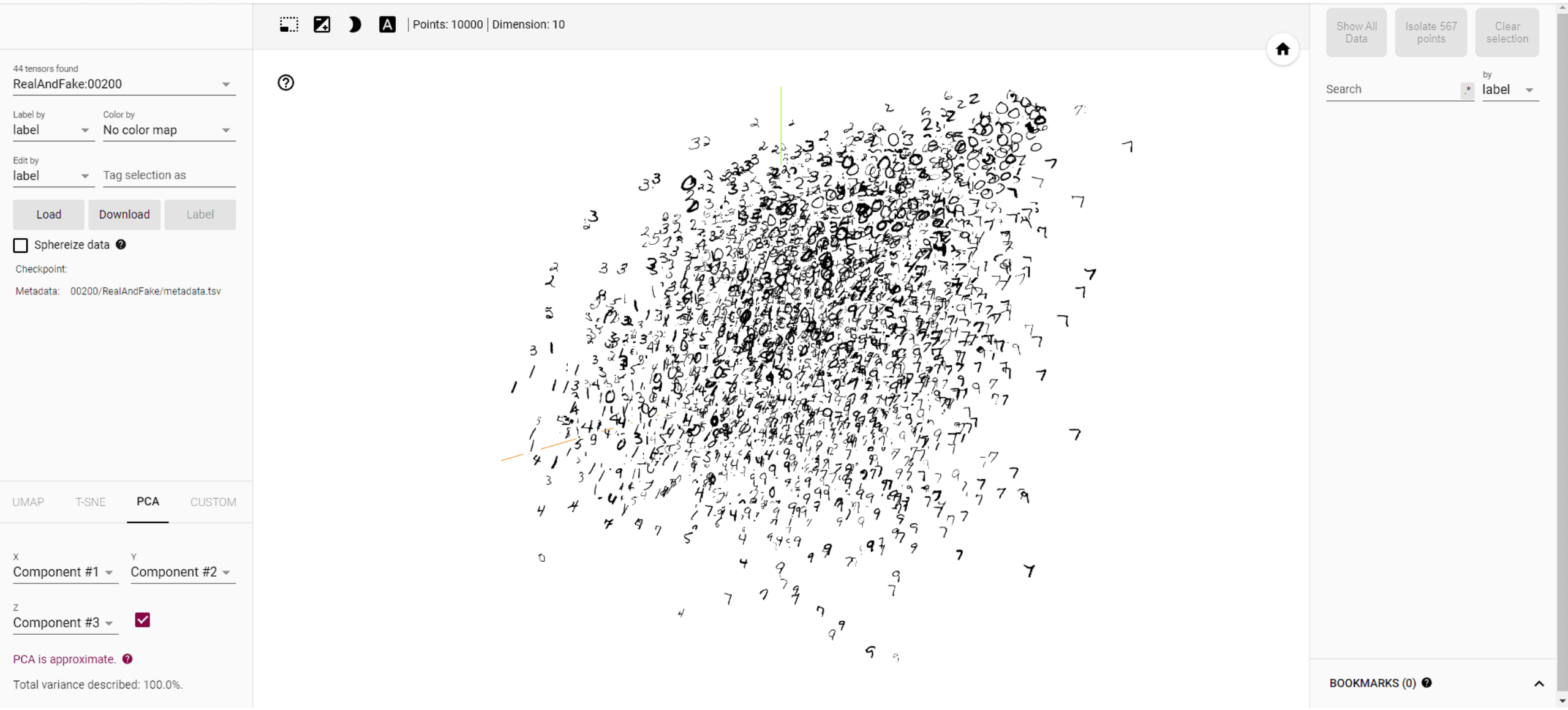
Results

Visualize & Targeting with MVM+ !



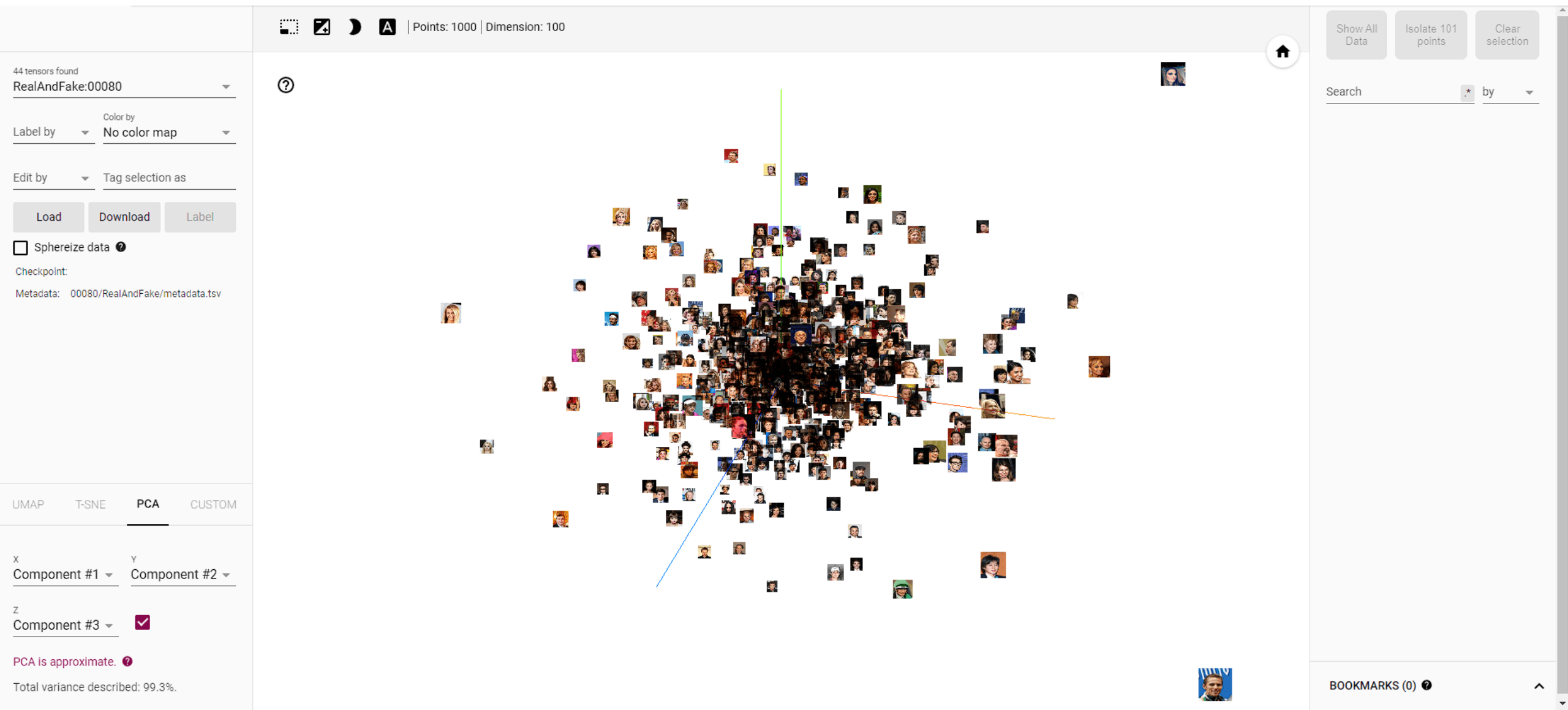
Results

Visualize & Targeting with MVM+ !



Results

Visualize & Targeting with MVM+ !



Conclusion

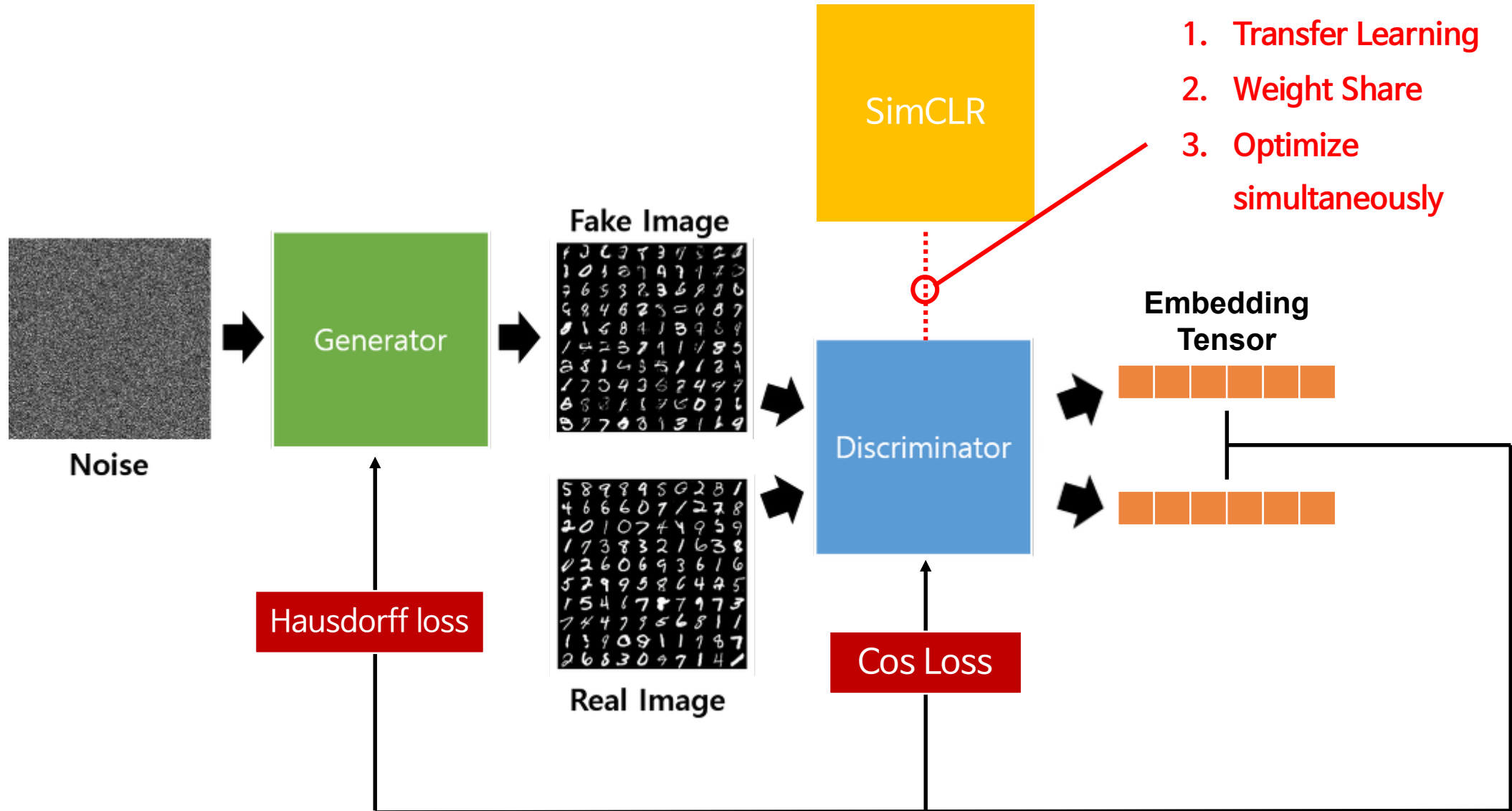
- 통계적 관점이 아닌, 기하학적인 관점에서 데이터를 표현하고 생성
- 기존 MVM의 부족한 부분을 보완, 안정성과 생성 능력 상승
- 학습 결과로 생성 데이터만이 아닌, 2~100 차원의 임베딩을 생성
 - 샘플 데이터셋이 데이터 공간에서 어떤 모양을 갖는지 **시각화**
 - 데이터간의 **유사성**을 표현 (가까운 데이터와 먼 데이터)
 - 특정 샘플 데이터에 가까운 생성 데이터를 **타겟팅**하고 생성할 수 있다.

Future Works

- GAN 과 MVM+의 결합
 - SOTA GAN + (MVM+)
- Self-supervised contrastive learning + (MVM+)
 - SimCLR + (MVM+)
 - BYOL + (MVM+)

Conclusion

Future Works



Thank you !