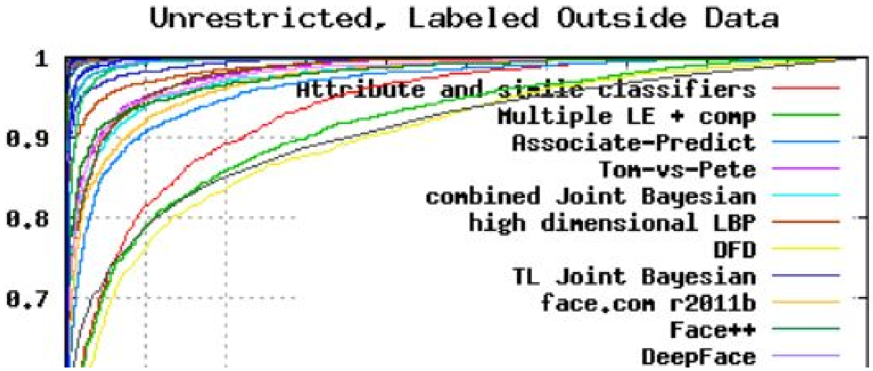
Modeling Distributions in Computer Vision

Haoqiang Fan
fhq@megvii.com

Face++ 旷视

About me

Joined Megvii (Face++) since 2012



Today’s topic

How to **neurally** process an unordered set?

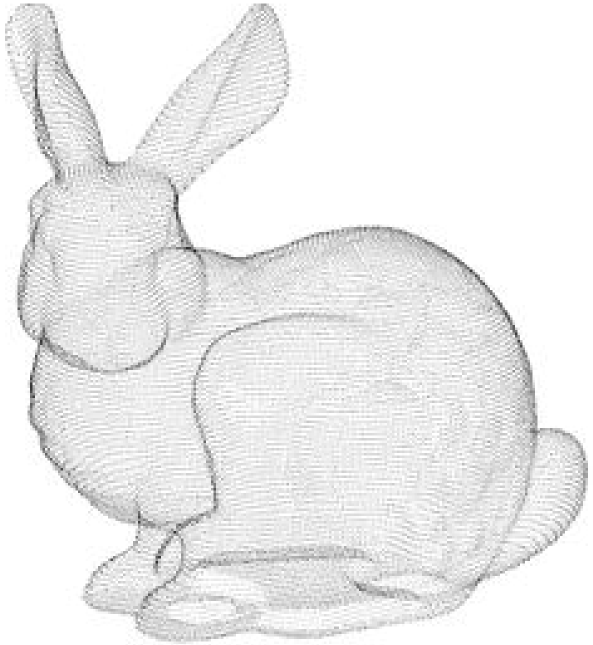
Generation

Evaluation

Example: 3D Deep Learning

Say we want to learn a “point cloud”

Crux: neural networks training needs an order!



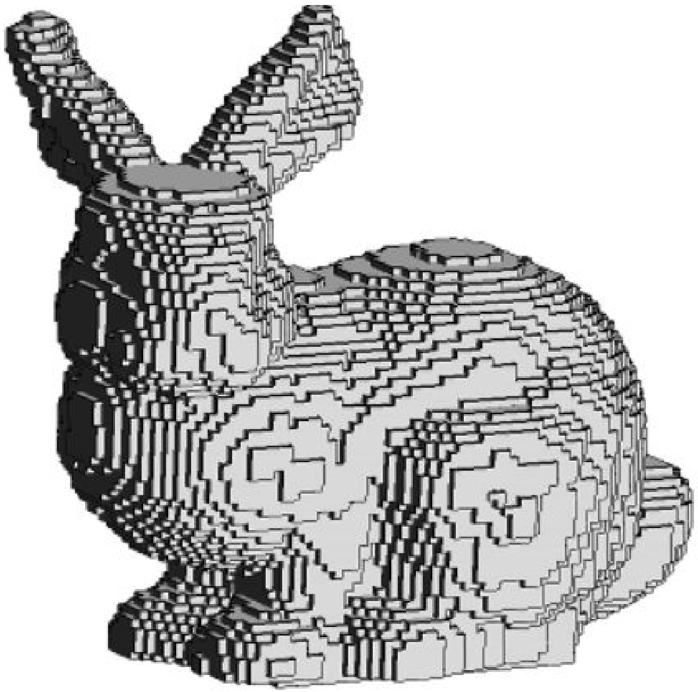
Comparison of representations

The volumetric method:

f(input) -> array of size S*S*S

issue of efficiency: only O(S*S) "interesting" cells

Hard to scale to high dim.



Comparison of representations

The point-based method

f(input) -> array of (x,y,z) triples

fixed length v.s. variable length (e.g. RNN)

need to deal with permutation

Comparison of representations

Continuous version:

volumetric f(input, x, y, z) -> p “energy” “discriminator”

issue: how to sample?

point-based f(input, r) -. (x,y,z) “decoder” “generator”

issue: how to estimate density?

Application: single image 3D reconstruction

Given one RGB image, predict the whole 3D shape of the designated object.

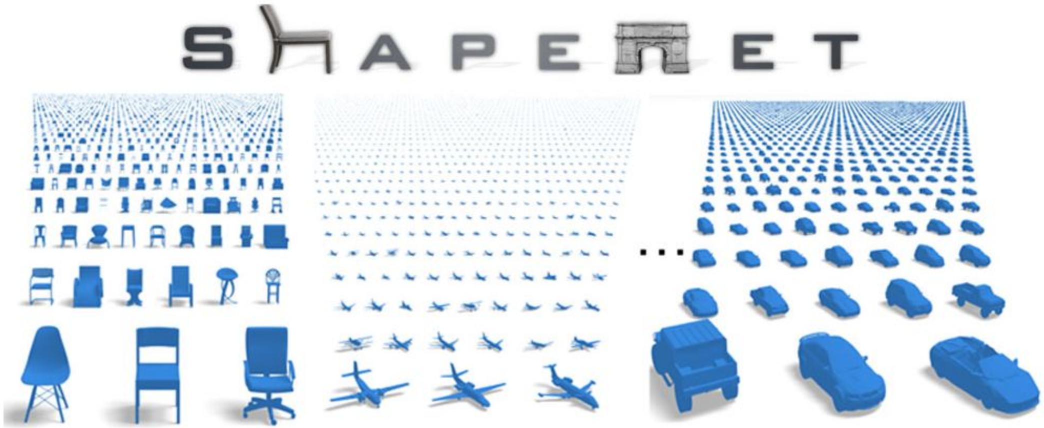
Point-cloud based representation

CVPR 2017 (oral)

Input

Reconstructed 3D point cloud

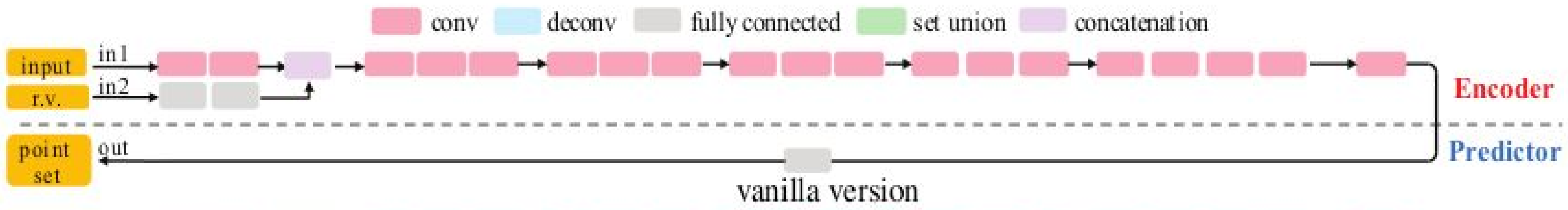
Training by synthetic data



Face++ 旷视

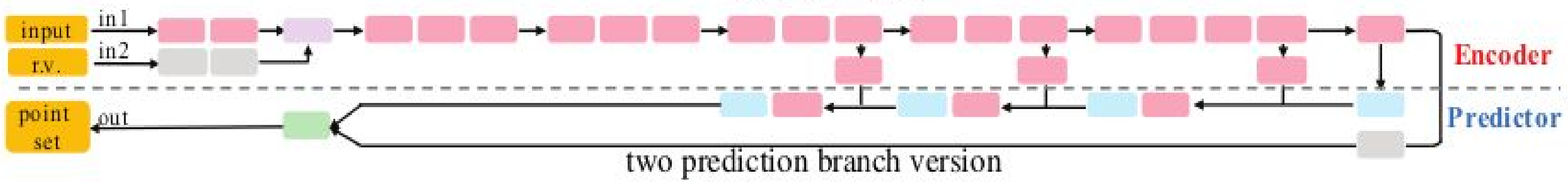
Network structure

image in, 1024 * 3 out



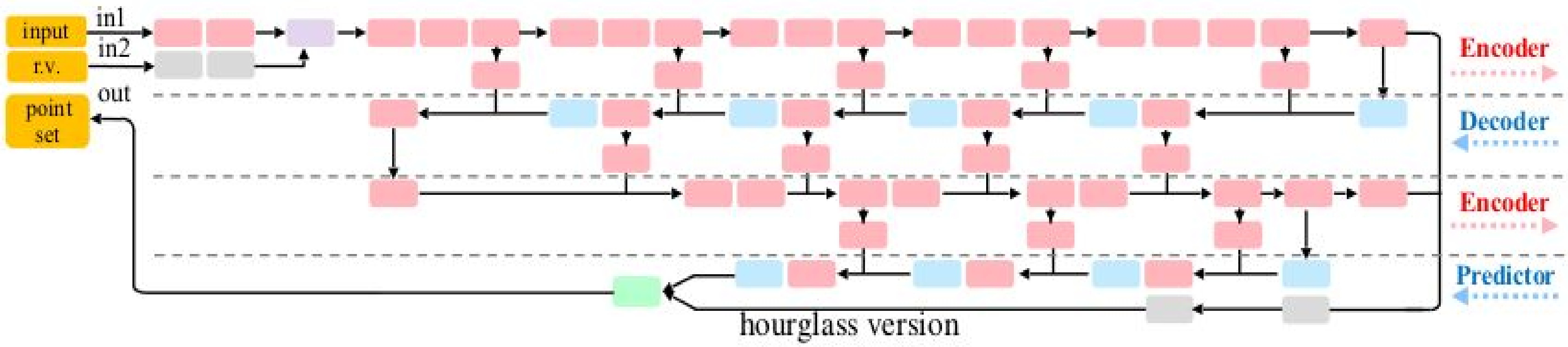
Network structure

image in, 1024 * 3 out

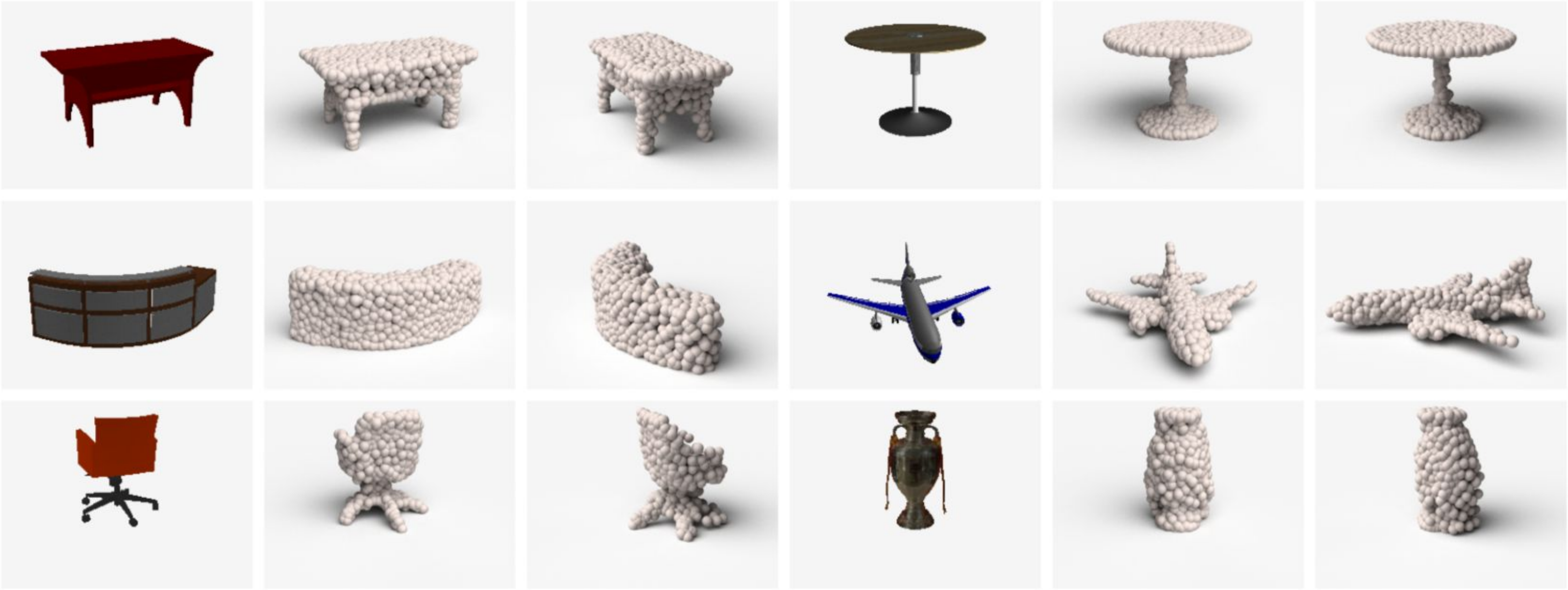


Network structure

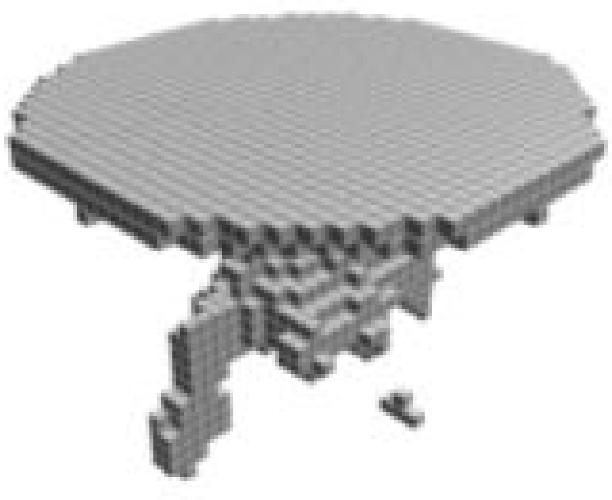
image in, 1024 * 3 out



Result: Synthetic Data

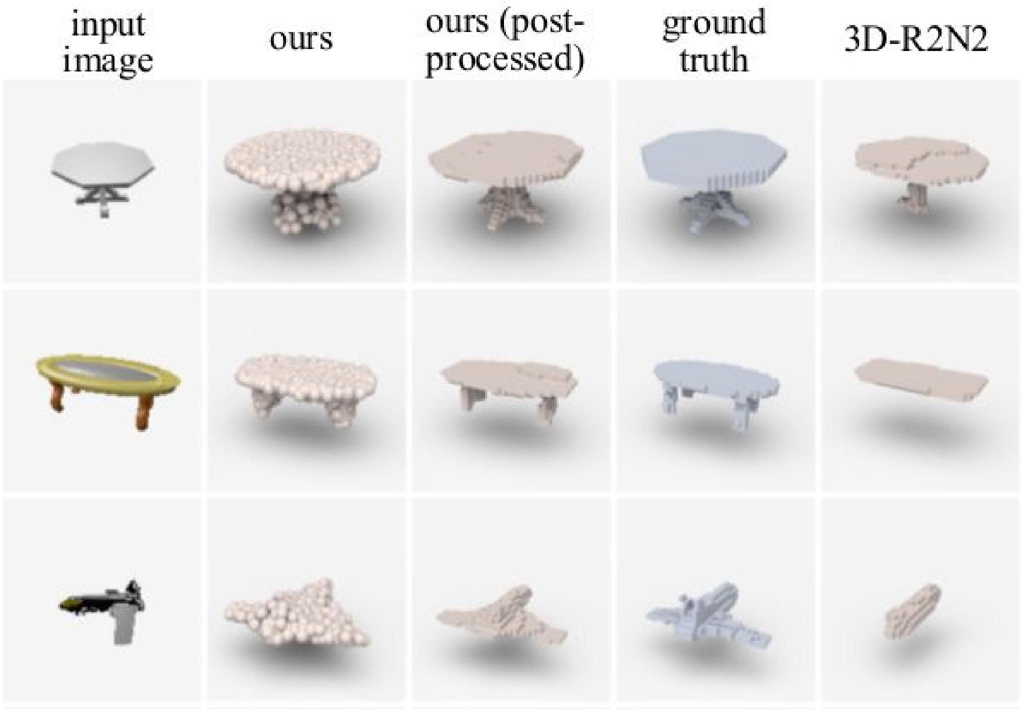


Comparison: previous state-of-the-art



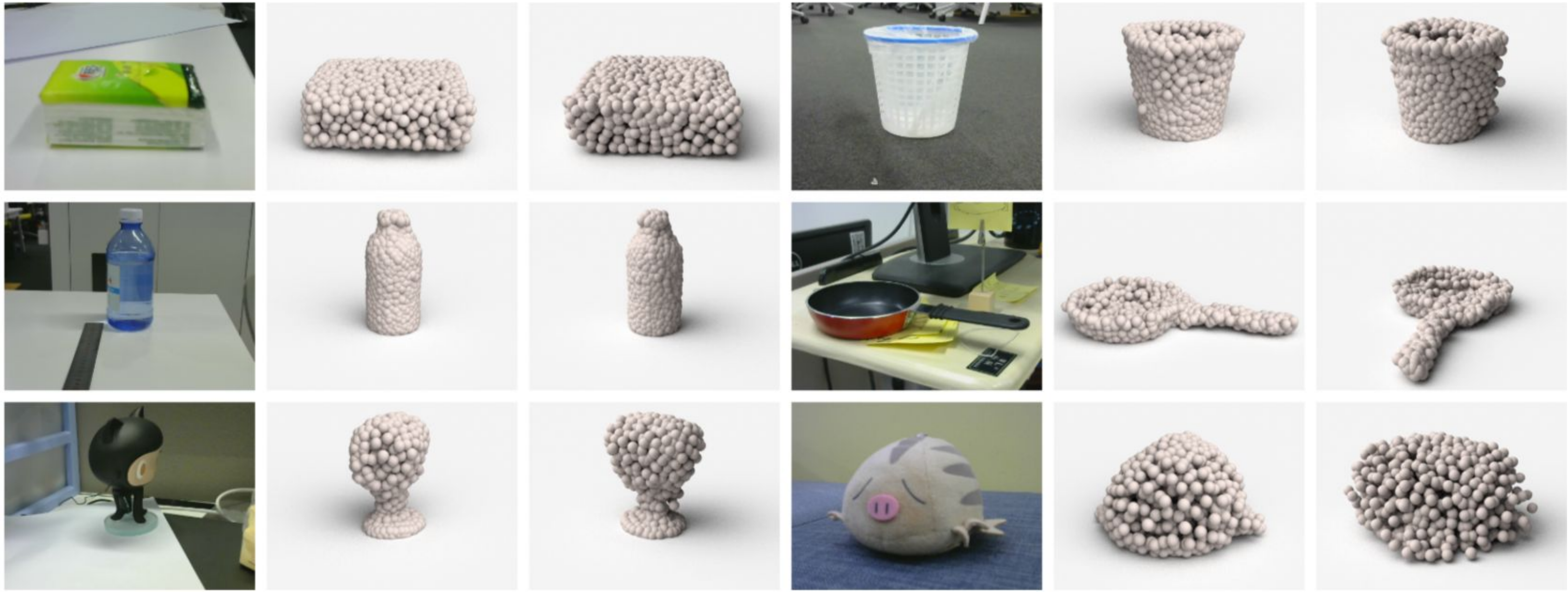
3D-R2N2 A Unified Approach for Single and Multi-view 3D Object Reconstruction   Christopher et. al.

Comparison to previous work



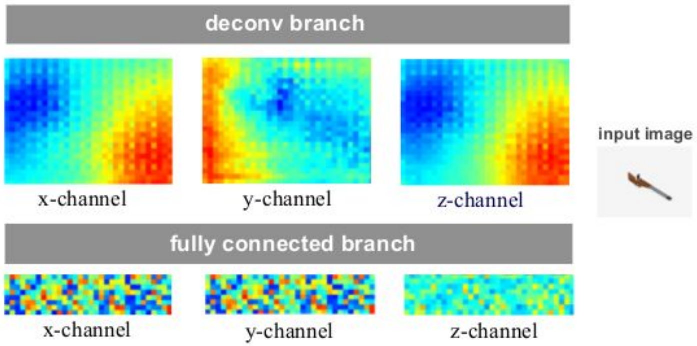
category	Ours	3D-R2N2		
	1 view	1 view	3 views	5 views
plane	**0.601**	0.513	0.549	0.561
bench	**0.550**	0.421	0.502	0.527
cabinet	0.771	0.716	0.763	**0.772**
car	0.831	0.798	0.829	**0.836**
chair	0.544	0.466	0.533	**0.550**
monitor	0.552	0.468	0.545	**0.565**
lamp	**0.462**	0.381	0.415	0.421
speaker	**0.737**	0.662	0.708	0.717
firearm	**0.604**	0.544	0.593	0.600
couch	**0.708**	0.628	0.690	0.706
table	**0.606**	0.513	0.564	0.580
cellphone	0.749	0.661	0.732	**0.754**
watercraft	**0.611**	0.513	0.596	0.610
mean	**0.640**	0.560	0.617	0.631

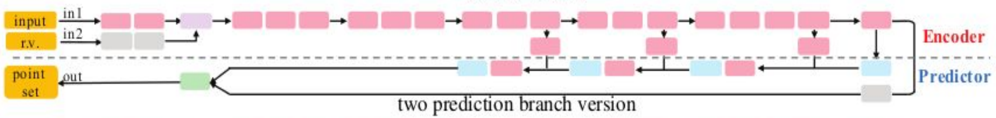
Result: Real Data



What does the network learn?

a **neural** ordering of the points!

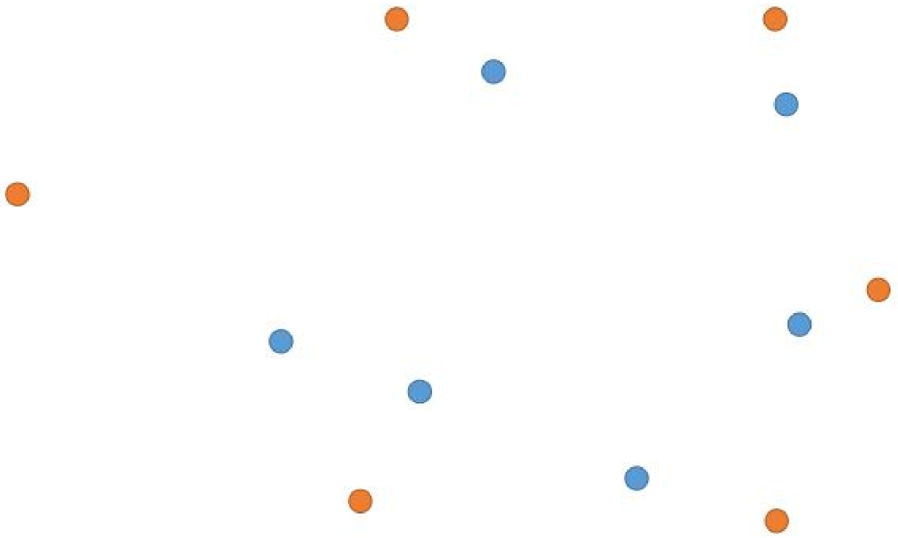




Loss function

How to compare two point distributions?

Need to be differentiable and permutation invariant.

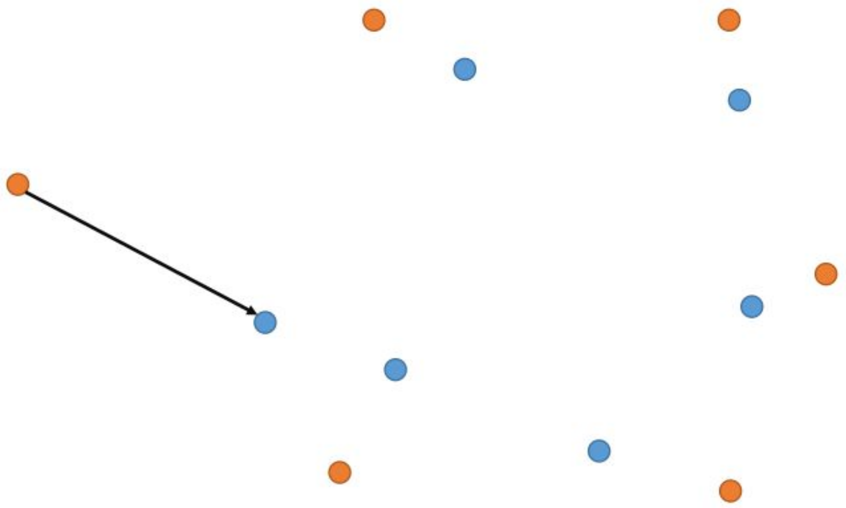


Loss function

Hausdorff Distance:

$$d_{\mathrm{HD}}(S_1, S_2) = \max\{\max_{x_i \in S_1} \min_{y_j \in S_2} \|x_i - y_j\|, \max_{y_j \in S_2} \min_{x_i \in S_1} \|x_i - y_j\|\}$$

problem: slow to optimize



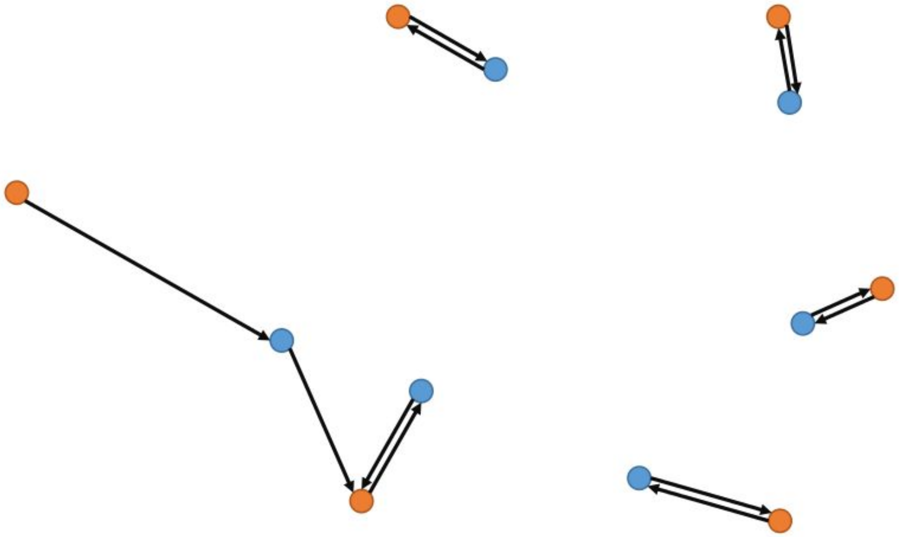
Face++ 旷视

Loss function

Symmetric Chamfer Distance:

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$

problem: unmatched density



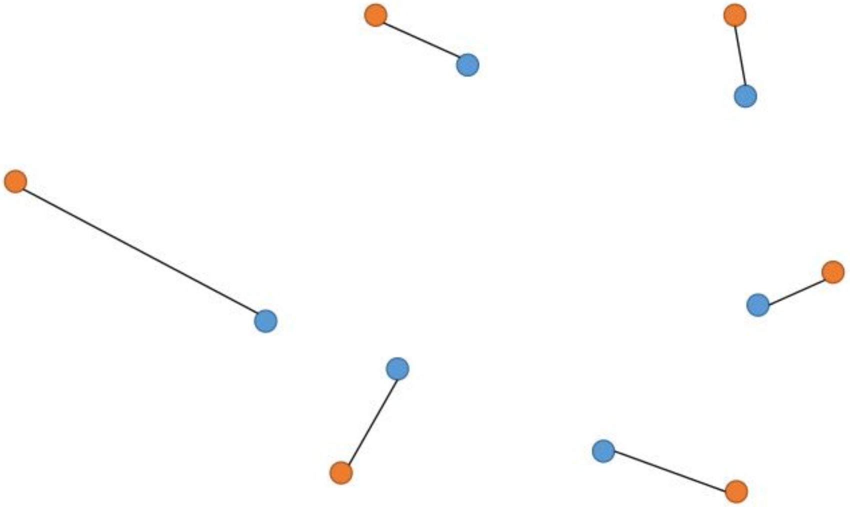
Face++ 旷视

Loss function

Earth Mover Distance

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

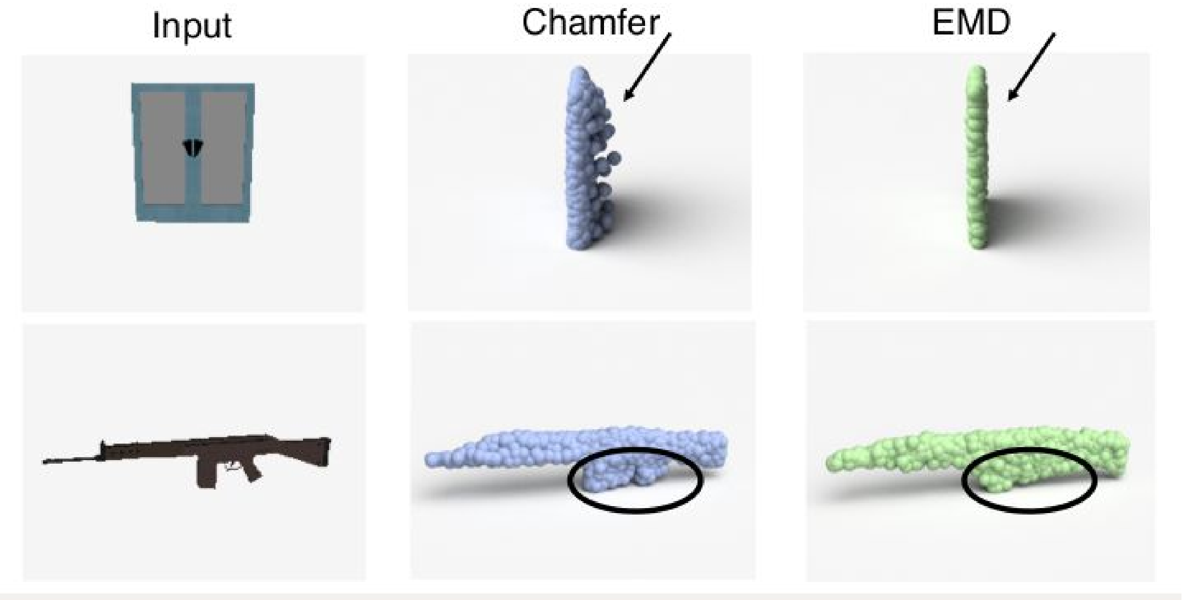
problem: slower to compute



Face++ 旷视

The “mean shape” behavior

Choice of loss function affects prediction

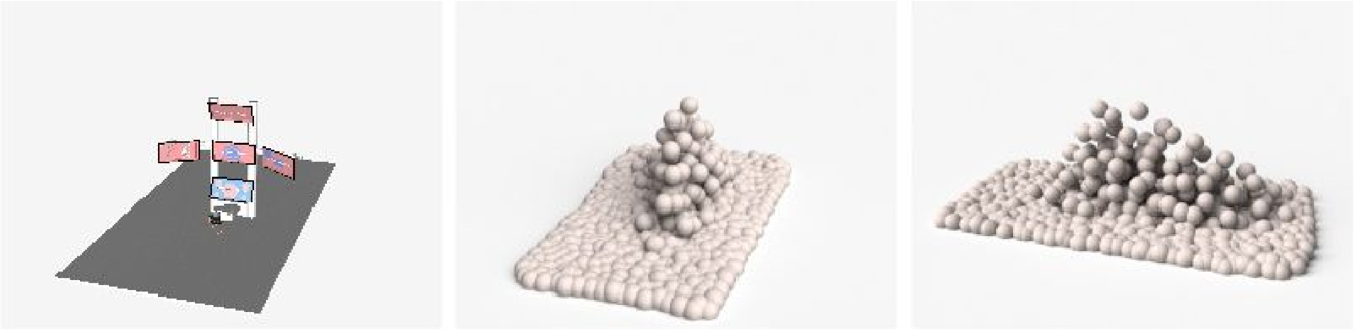


Face++ 旷视

The “mean shape” behavior

The crux: the output minimizes the distance to all “unresolvable” shapes

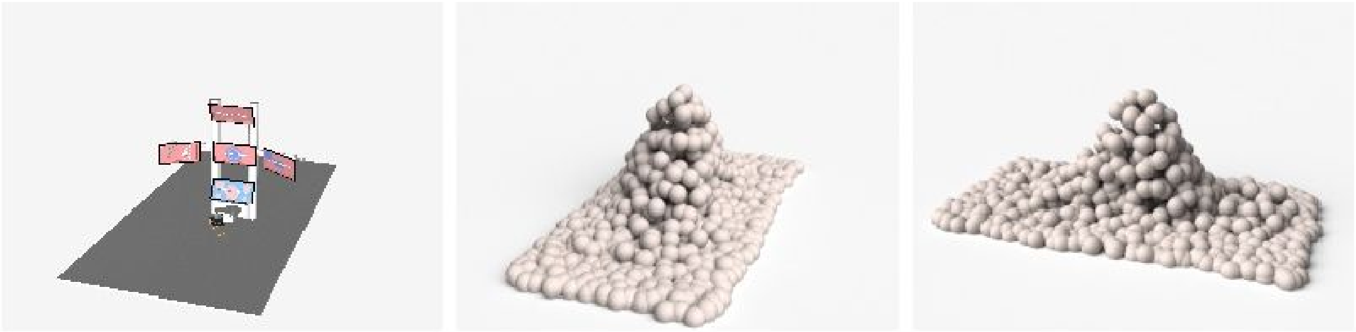
due to inherent ambiguity of single view, or limited network capacity.



symmetric chamfer distance

The “mean shape” behavior

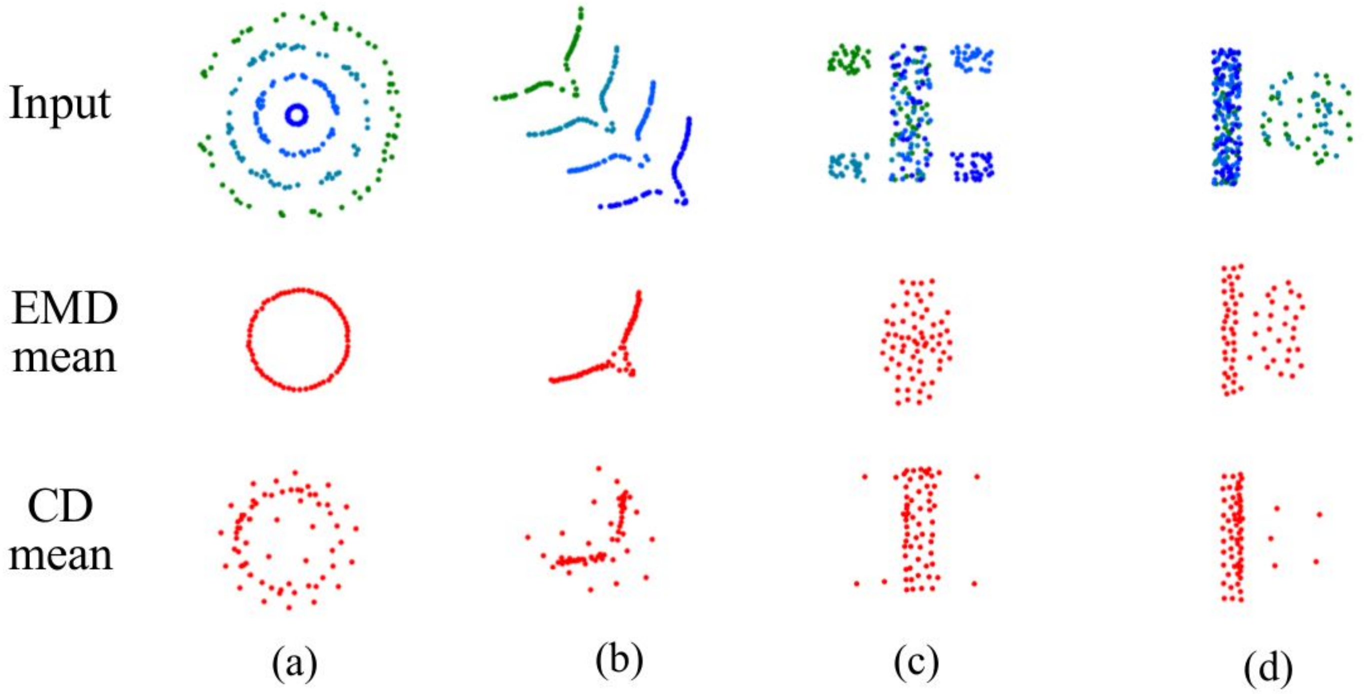
The crux: the output minimizes the distance to all “unresolvable” shapes due to inherent ambiguity of single view, or limited network capacity.



earth mover distance

Face++ 旷视

Synthetic experiment



How to model uncertainty?

Distribution of distribution

The second distribution: point-based, earth mover distance.

The first distribution: ??

Distributional modeling methods

VAE

?GAN

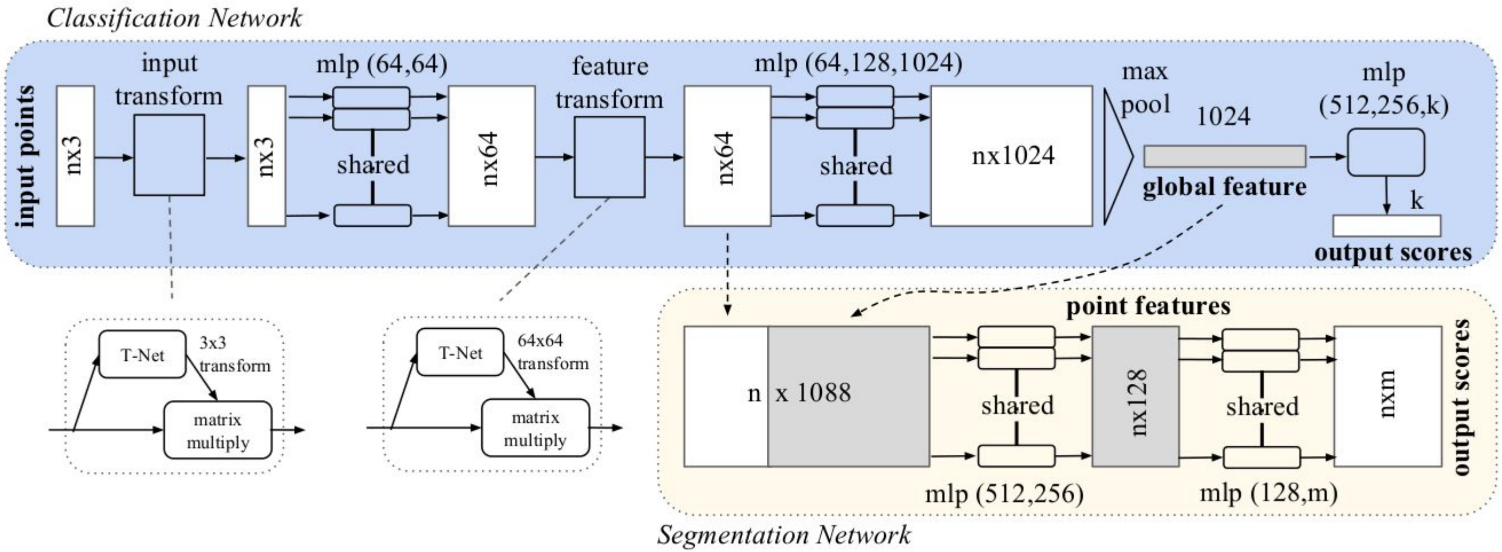
The problem of GAN

We need a discriminator that takes input from target domain.

This is sometimes non-trivial: how to neurally process a point set?

The problem of GAN

deep symmetric functions



Face++ 旷视

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation Hao et. al.

Streaming algorithms

Say if you want to count the number of distinct elements in a set.

Process one item at a time, only constant working memory.

Solution: 1/min(h(x) for x in X)

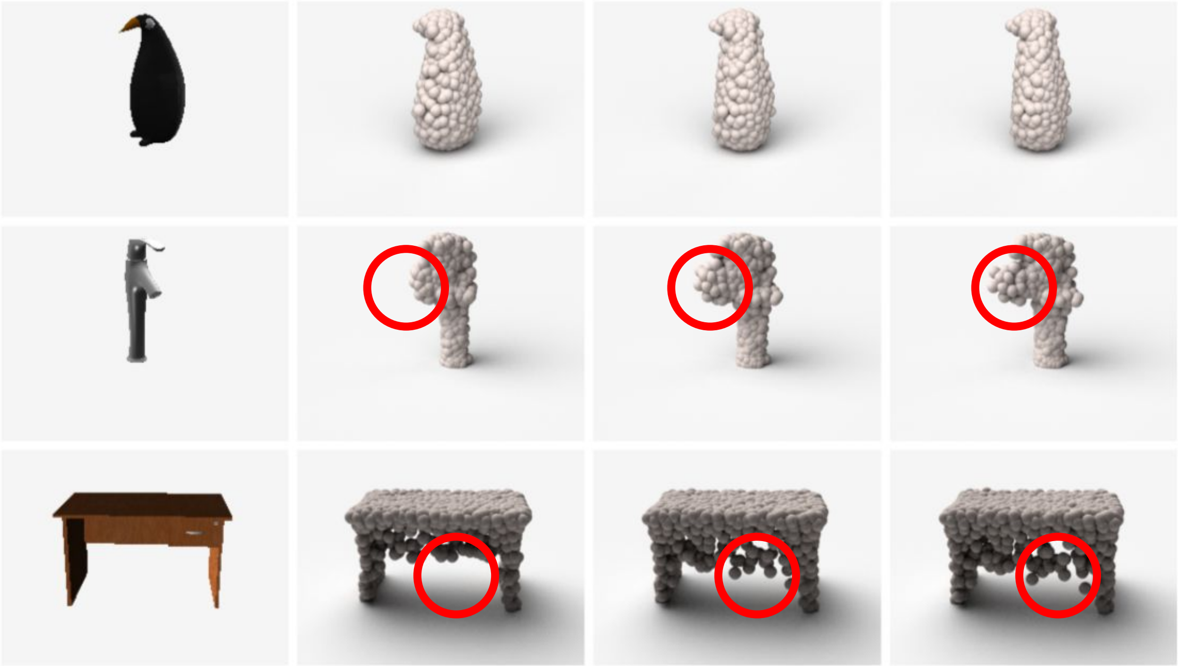
h: X -> [0,1] a random hash function

What if we do not want to use GAN

We can “encourage” the network to spread its points:

min(loss(label,f(input,r1)),loss(label,f(input,r2)))

Simpler solution?



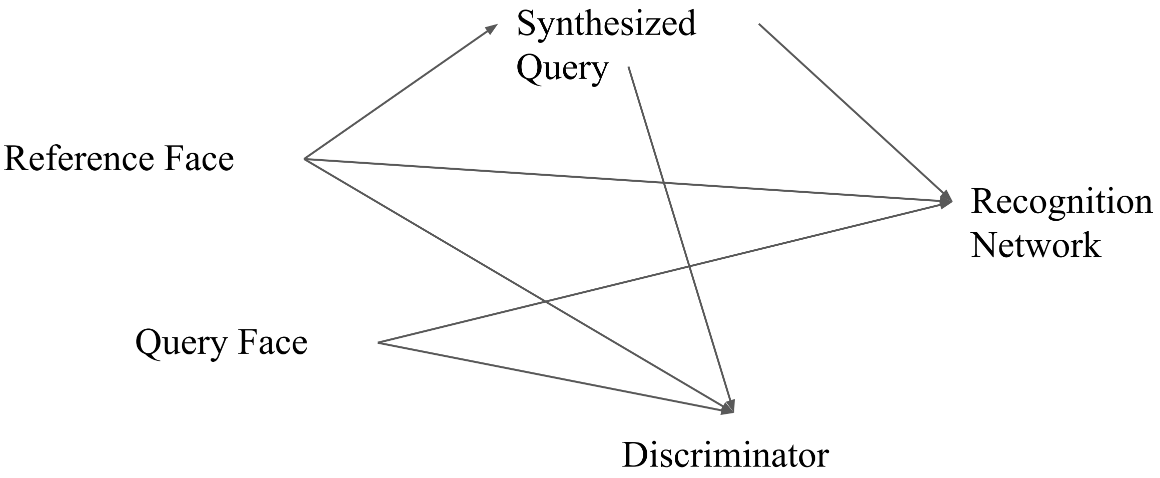
Face++ 旷视

More on distributional modeling

GAN in practice

Can we use GAN to improve face recognition?

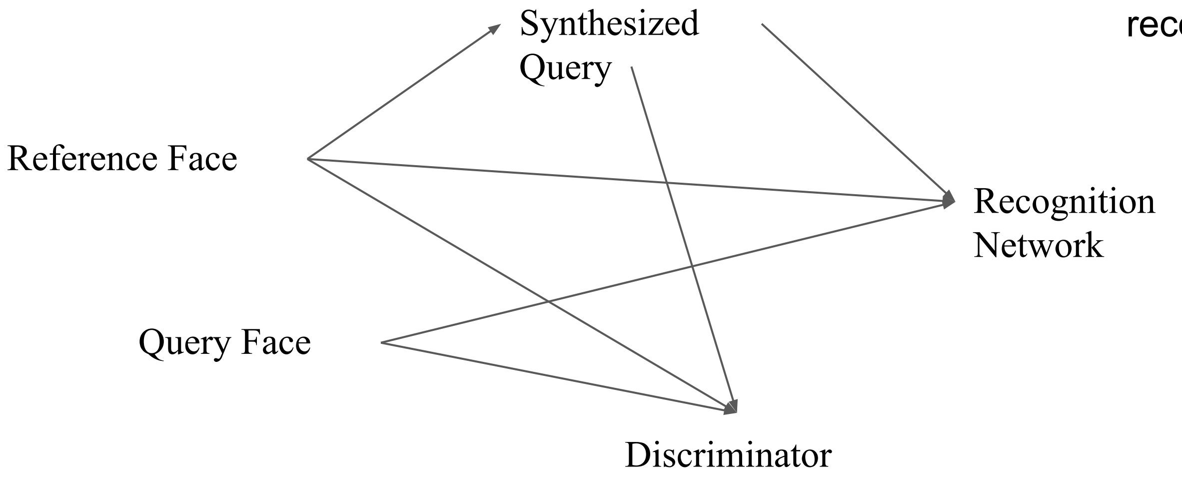
Training by generated data?



Face++ 旷视

Can we use GAN to improve face recognition?

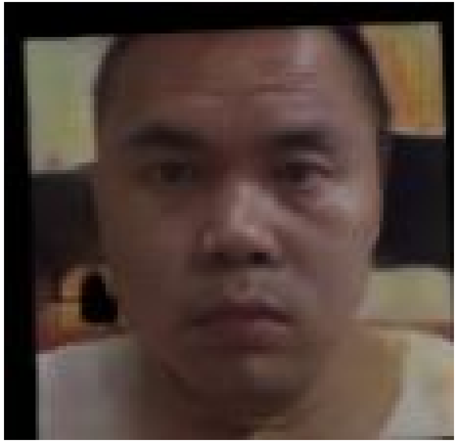
Training by generated data?



Problem:
Discrimination is no easier than recognition.

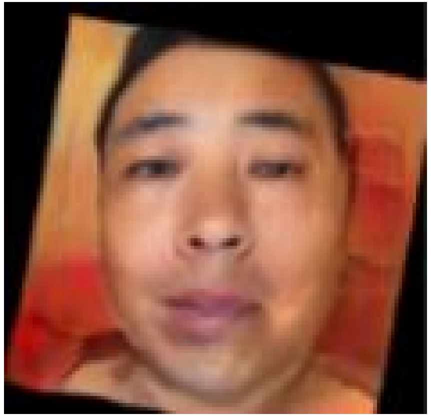
Can we use GAN to improve face recognition?

What if we remove the identity preserving loss?



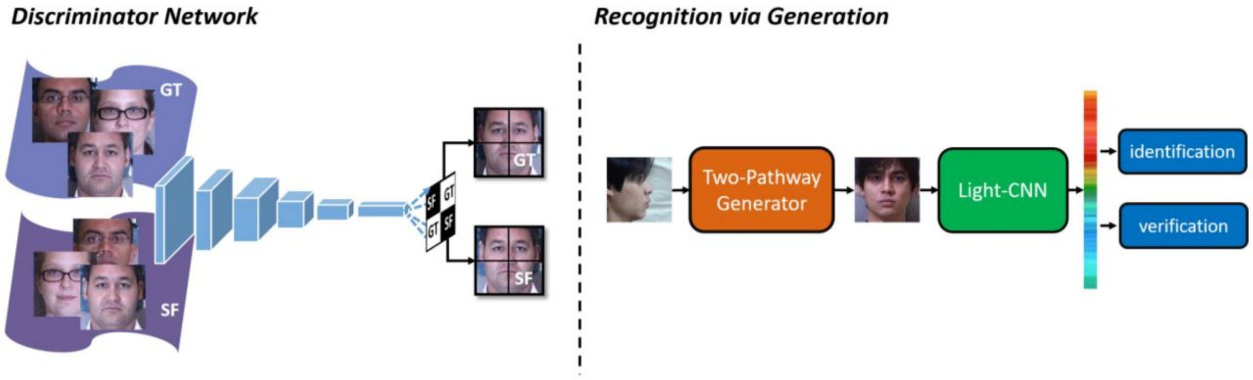
Can we use GAN to improve face recognition?

Then it may not preserve identity!



Can we use GAN to improve face recognition?

Solution 1: adversarial loss in pre-processing stage

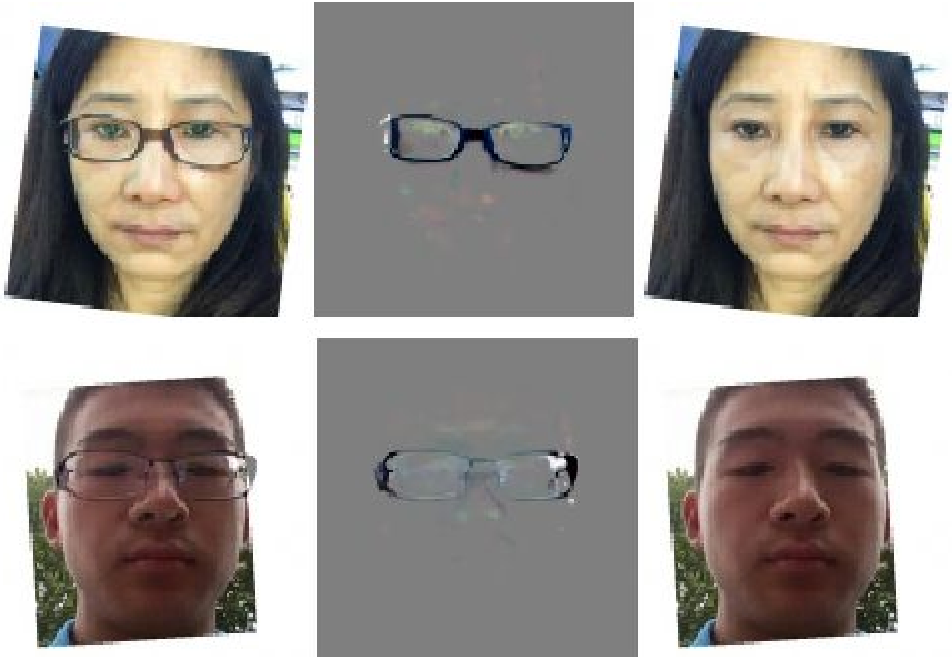


Beyond Face Rotation Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis   Rui et. al.

Face++ 旷视

Can we use GAN to improve face recognition?

Solution 2: GAN based data augmentation



Face++ 旷视

Can we use GAN to improve face recognition?

Solution 2: GAN based data augmentation

Face++ 旷视

Conclusion

Modeling a distribution: metric between distributions

How you measure the distance influences what you learn.

Career@megvii.com